
Kerapu Documentation

P.R. Water

okt. 26, 2020

Inhoud:

1	Installatie	3
1.1	Voorbereiding	3
2	Voorbeeld	5
3	Testset	7
4	Licentie	9
5	API	11
5.1	kerapu package	11
	Python-moduleindex	33
	Index	35

Een implementatie van de Grouper in Python.

HOOFDSTUK 1

Installatie

Kerapu kan eenvoudig geïnstalleerd worden met pip:

```
pip3 install kerapu
```

1.1 Voorbereiding

Alvorens gebruik te kunnen maken van Kerapu moeten de boombestanden en referentietabellen worden geconverteerd en opgeslagen. Ten tijde van schrijven van dit document was de meest recente versie van Grouper Tabellen v20180920, vervang in de onderstaande tekst deze versie voor de meest recente versie.

- Download het bestand Grouper Tabellen v20180920 van https://puc.overheid.nl/nza/doc/PUC_259930_22/.
- Extract de XML-bestanden uit het ZIP-bestand:

```
unzip -x "20190101_Groupertabellen_v20180920.zip"
```

- Converteer de XML-bestanden naar CSV (in het voorbeeld hieronder worden de CSV-bestanden weggeschreven in de folder var/lib):

```
kerapu kerapu:shredder "20190101_BoomBestanden_v20180920.xml" var/lib/
kerapu kerapu:shredder "20190101_Riferenties_v20180920.xml" var/lib/
```


HOOFDSTUK 2

Voorbeeld

Hieronder een voorbeeld om de zorgproductcode van een subraject af te leiden.

```
from kerapu.Kerapu import Kerapu
from kerapu.lbz.Subraject import Subraject

# Maak een Grouper object en laad boombestanden en referentietabellen.
grouper = Kerapu()
grouper.init_static('var/lib')

# Maak een subraject object.
subraject = Subraject('1',                               # Subrajectnummer
                      '0303',                           # Zorgverlenerspecificatiecode
                      '0280',                           # Diagnosecode
                      '11',                             # Zorgtypecode
                      '000',                            # Zorgvraagcode
                      '2012-01-01',                     # Begindatum subraject
                      '2000-01-01',                     # Geboortedatum
                      'M',                             # Geslachtscode
                      '01234567')                      # AGB-code zorginstelling

# Voeg uitgevoerde zorgactiviteiten aan het subraject toe.
subraject.add_zorg_activiteit('038940', 1)
subraject.add_zorg_activiteit('038941', 1)
subraject.add_zorg_activiteit('190012', 1)
subraject.add_zorg_activiteit('190015', 1)

# Bepaal zorgproductgroep en zorgproduct.
zorg_product_code = grouper.bepaal_zorg_product(subraject)

print('Zorgproductgroepcode: {}'.format(subraject.zorg_product_groep_code))
print('Zorgproductcode: {}'.format(subraject.zorg_product_code))
```


HOOFDSTUK 3

Testset

De grouperbestanden bevatten sinds 2019 een testset. Deze testset is terug te vinden in bestand `var/lib/testset.csv` en zijn een onderdeel van de unittesten van Kerapu.

Het commando voor het converteren van de XML-bestanden met test data is:

```
./bin/kerapu kerapu:test-shredder ~/Downloads/20190101\ Testset\ Grouper\ RZ19b\_\  
→v20180920.zip test/var/lib/testset.csv
```


HOOFDSTUK 4

Licentie

Dit project is gelicentieerd onder de [MIT-licentie](#).

5.1 kerapu package

5.1.1 Subpackages

kerapu.application package

Submodules

kerapu.application.KerapuApplication module

```
class kerapu.application.KerapuApplication.KerapuApplication
    Basisklassen: cleo.application.Application
```

The Kerapu application.

```
get_default_commands() → List[cleo.commands.command.Command]
    Returns the default commands of this application.
```

Return type list[Command]

kerapu.application.kerapu module

```
kerapu.application.kerapu.main() → None
    Entry point for the kerapu console script.
```

Module contents

kerapu.boom package

Subpackages

kerapu.boom.attribuut package

Submodules

kerapu.boom.attribuut.Attribuut module

```
class kerapu.boom.attribuut.Attribuut(attribuut_id: int,
                                         boom_parameter_nummer: int)
```

Basisklassen: `object`

Abstract klasse voor attributen.

```
abstract tel(subraject: kerapu.lbz.Subraject.Subraject) → int
```

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters `subraject` (`Subraject`) – Het subraject.

Return type `int`

kerapu.boom.attribuut.Attribuut12 module

```
class kerapu.boom.attribuut.Attribuut12(attribuut_id: int,
                                         boom_parameter_nummer: int,
                                         filter_waarde: str)
```

Basisklassen: `kerapu.boom.attribuut.Attribuut`

Klasse voor attributen met toetswijze 1 (gelijk) en waarde type 2 (alfanumeriek).

```
tel(subraject: kerapu.lbz.Subraject.Subraject) → int
```

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters `subraject` (`Subraject`) – Het subraject.

Return type `int`

kerapu.boom.attribuut.Attribuut21 module

```
class kerapu.boom.attribuut.Attribuut21(attribuut_id: int,
                                         boom_parameter_nummer: int,
                                         onder_filter_waarde: int,
                                         boven_filter_waarde: int)
```

Basisklassen: `kerapu.boom.attribuut.Attribuut`

Klasse voor attributen met toetswijze 2 (tussen) en waarde type 1 (numeriek).

```
tel(subraject: kerapu.lbz.Subraject.Subraject) → int
```

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters `subraject` (`Subraject`) – Het subraject.

Return type `int`

Module contents

```
kerapu.boom.attribuut.maak_attribuut(attribuut_id: int, boom_parameter_nummer: int, filter_toets_wijze: int, filter_waarde_type: int, onder_filter_waarde: str, boven_filter_waarde: str) → kerapu.boom.attribuut.Attribuut
```

Fabriek voor het maken van attributen.

Parameters

- **attribuut_id** (*int*) – Het ID van het attribuut.
- **boom_parameter_nummer** (*int*) – Het ID van de boomparameter van het attribuut.
- **filter_toets_wijze** (*int*) – De filtertoetswijze
- **filter_waarde_type** (*int*) – Het type van de waarde van het attribuut.
- **onder_filter_waarde** (*str*) – De ondergrens.
- **boven_filter_waarde** (*str*) – De bovengrens.

Return type *Attribuut*

kerapu.boom.attribuut_groep_koppeling package

Submodules

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling module

```
class kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling
```

Basisklassen: *object*

Abstract klasse voor attribuutgroepkoppelingen.

abstract test (*subtraject*: kerapu.lbz.Subtraject.Subtraject) → *bool*
Test of een subtraject voldoet aan een attribuutgroepkoppeling.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type *bool*

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2 module

```
class kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2.
```

Basisklassen: `kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling`, `AttribuutGroepKoppeling`

Klasse voor attribuutgroepkoppelingen met filtertoetswijze 2 (tussen onder- en bovengrens).

test (`subtraject: kerapu.lbz.Subtraject.Subtraject`) → `bool`

Test of een subtraject voldoet aan een attribuutgroepkoppeling.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `bool`

Module contents

```
kerapu.boom.attribuut_groep_koppeling.maak_attribuut_groep_koppeling(attribute_groep_id:  
    int, at-  
    tribuut:  
    kerapu.boom.attribuut.Attribuut  
    attri-  
    buut_toets_wijze:  
    int, on-  
    der_toets_waarde:  
    int, bo-  
    ven_toets_waarde:  
    int) →  
    kerapu.boom.attribuut_groep_koppeling
```

Fabriek voor het maken van attribuutgroepkoppelingen.

Parameters

- **attribute_groep_id** (`int`) – Het ID van de koppeling.
- **attribuut** (`Attribuut`) – Het attribuut van de koppeling.
- **attribuut_toets_wijze** (`int`) – De attribuuttoetswijze.
- **onder_toets_waarde** (`int`) – De ondergrens.
- **boven_toets_waarde** (`int`) – De bovengrens.

Return type `AttribuutGroepKoppeling`

kerapu.boom.boom_parameter package**Submodules****kerapu.boom.boom_parameter.BehandelKlasse module**

```
class kerapu.boom.boom_parameter.BehandelKlasse(weeg_factor_nummer: int)
```

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter behandelklasse: som van aantal (met en zonder weegfactor).

Boomparameternummers: 351, 451, 551.

tel (*behandel_klasse_code: str, subraject: kerapu.lbz.Subraject.Subraject*) → *int*

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subraject voorkomt in een behandelklasse.

Parameters

- **behandel_klasse_code** (*str*) – De behandelklassecode waartegen getest moet worden.
- **subraject** (*Subraject*) – Het subraject.

Return type *int*

kerapu.boom.boom_parameter.BoomParameter module

```
class kerapu.boom.boom_parameter.BoomParameter.BoomParameter
```

Basisklassen: *object*

Abstarcte klasse voor boomparameters.

abstract tel (*waarde, subraject: kerapu.lbz.Subraject.Subraject*) → *int*

Geeft het aantal malen dat de boomparameter voldoet aan een waarde.

Parameters

- **waarde** ([*int / str*]) – De waarde waartegen getest moet worden.
- **subraject** (*kerapu.lbz.Subraject.Subraject*) – Het subraject.

Return type *int*

kerapu.boom.boom_parameter.DiagnoseCluster module

```
class kerapu.boom.boom_parameter.DiagnoseCluster.DiagnoseCluster(cluster_nummer: int)
```

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter diagnosecluster.

Boomparameternummers: 232, 233, 234, 235, 236, 237.

tel (*cluster_code: str, subraject: kerapu.lbz.Subraject.Subraject*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat een subraject voldoet aan een diagnoseclustercode.

Parameters

- **cluster_code** (*str*) – De cluster_code waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type *int*

kerapu.boom.boom_parameter.DiagnoseCode module

```
class kerapu.boom.boom_parameter.DiagnoseCode.DiagnoseCode
Basisklassen: kerapu.boom.boom_parameter.BoomParameter.BoomParameter

Klasse voor boomparameter diagnosecode.

Boomparameternummer: 230.

tel (diagnose_attribuut_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat een subtraject voldoet aan een een (specialismecode, diagnosecode) combinatie.
```

Parameters

- **diagnose_attribuut_code** (*str*) – De attribuutcode voor (specialismecode, diagnosecode) combinatie.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type *int*

kerapu.boom.boom_parameter.Geslacht module

```
class kerapu.boom.boom_parameter.Geslacht.Geslacht
Basisklassen: kerapu.boom.boom_parameter.BoomParameter.BoomParameter

Klasse voor boomparameter geslacht.

Boomparameternummer: 230.

tel (geslacht_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat de patient van een subtraject voldoet aan een geslacht.
```

Parameters

- **geslacht_code** (*str*) – De geslachtscode waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type *int*

kerapu.boom.boom_parameter.Leeftijd module

```
class kerapu.boom.boom_parameter.Leeftijd.Leeftijd
Basisklassen: kerapu.boom.boom_parameter.BoomParameter.BoomParameter

Klasse voor boomparameter leeftijd.

Boomparameternummer: 100.

tel (leeg, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
Geeft de leeftijd van de patient van een subtraject.
```

Parameters

- **leeg** (*None*) – Wordt niet gebruikt.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

`kerapu.boom.boom_parameter.SpecialismeCluster module`

```
class kerapu.boom.boom_parameter.SpecialismeCluster.SpecialismeCluster(cluster_nummer:  
int)
```

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter specialismecluster.

Boomparameternummers: 210, 202.

tel (cluster_code: `str`, subtraject: `kerapu.lbz.Subtraject.Subtraject`) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van een subtraject voldoet aan een specialismecluster.

Parameters

- **cluster_code** (`str`) – De clustercode waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

`kerapu.boom.boom_parameter.SpecialismeCode module`

```
class kerapu.boom.boom_parameter.SpecialismeCode.SpecialismeCode
```

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter specialismecode.

Boomparameternummer: 200.

tel (specialisme_code: `str`, subtraject: `kerapu.lbz.Subtraject.Subtraject`) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van een subtraject voldoet aan een specialismecode.

Parameters

- **specialisme_code** (`str`) – De specialismecode.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

`kerapu.boom.boom_parameter.ZorgActiviteitCluster module`

```
class kerapu.boom.boom_parameter.ZorgActiviteitCluster.ZorgActiviteitCluster(cluster_nummer:  
int,  
weeg_factor_nummer:  
int)
```

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter zorgactiviteitcluster.

Boomparameternummers: 301..310, 401..410, 401..510.

tel (*cluster_code: str, subtraject: kerapu.lbz.Subraject.Subraject*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subraject voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **subraject** (*Subraject*) – Het subraject.

Return type int

kerapu.boom.boom_parameter.ZorgActiviteitCode module

class kerapu.boom.boom_parameter.ZorgActiviteitCode (*weeg_factor_nummer: int*)

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter zorgactiviteit.

Boomparameternummers: 300, 400, 500.

tel (*zorg_activiteit_code: str, subraject: kerapu.lbz.Subraject.Subraject*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subraject voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **subraject** (*Subraject*) – Het subraject.

Return type int

kerapu.boom.boom_parameter.ZorgInstellingCode module

class kerapu.boom.boom_parameter.ZorgInstellingCode (**ZorgInstellingCode**)

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter zorginstelling.

Boomparameternummer: 110.

tel (*zorg_instelling_code: str, subraject: kerapu.lbz.Subraject.Subraject*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat het zorginstelling van een subraject voldoet een zorginstelling-code.

Parameters

- **zorg_instelling_code** (*str*) – De AGB-code waaraan de zorginstelling moet voldoen.
- **subraject** (*Subraject*) – Het subraject.

Return type int

kerapu.boom.boom_parameter.ZorgTypeCode module

```
class kerapu.boom.boom_parameter.ZorgTypeCode.ZorgTypeCode
    Basisklassen: kerapu.boom.boom\_parameter.BoomParameter.BoomParameter

    Klasse voor boomparameter zorgtype.

    Boomparameternummer: 210.

    tel (zorg_type_attribuut_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
        Geeft het aantal malen (d.w.z. 0 of 1) dat het zorgtype van een subtraject voldoet aan een (specialismecode, zorgtypecode) combinatie.
```

Parameters

- **zorg_type_attribuut_code** ([str](#)) – De attribuutcode voor (specialismecode, zorgtypecode) combinatie.
- **subtraject** ([Subtraject](#)) – Het subtraject.

Return type [int](#)

kerapu.boom.boom_parameter.ZorgVraagCluster module

```
class kerapu.boom.boom_parameter.ZorgVraagCluster.ZorgVraagCluster (cluster_nummer: int)
    Basisklassen: kerapu.boom.boom\_parameter.BoomParameter.BoomParameter

    Klasse voor boomparameter zorgvraagcluster.

    Boomparameternummers: 221, 222.

    tel (cluster_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
        Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voorkomt in een zorgvraagcluster.
```

Parameters

- **cluster_code** ([str](#)) – De cluster_code waartegen getest moet worden.
- **subtraject** ([Subtraject](#)) – Het subtraject.

Return type [int](#)

kerapu.boom.boom_parameter.ZorgVraagCode module

```
class kerapu.boom.boom_parameter.ZorgVraagCode.ZorgVraagCode
    Basisklassen: kerapu.boom.boom\_parameter.BoomParameter.BoomParameter

    Klasse voor boomparameter zorgvraagcode.

    Boomparameternummers: 220.

    tel (zorg_vraag_attribuut_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject) → int
        Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voldoet aan een (specialismecode, zorgvraagcode).
```

Parameters

- **zorg_vraag_attribuut_code** ([str](#)) – De attribuutcode voor (specialismecode, zorgvraagcode).
- **subtraject** ([Subtraject](#)) – Het subtraject.

Return type `int`

Module contents

```
kerapu.boom.boom_parameter.create_boom_parameter(boom_parameter_nummer: int) →  
    kerapu.boom.boom_parameter.BoomParameter.BoomParamet
```

Een fabriek met hergebruik voor het maken van boomparameters. Het aanroepen van deze functie met hetzelfde boomparameternummer zal tevens hetzelfde object opleveren.

Parameters `boom_parameter_nummer` (`int`) – Het nummer van de boomparameter.

Return type `BoomParameter`

Submodules

kerapu.boom.AttribuutGroep module

```
class kerapu.boom.AttribuutGroep.AttribuutGroep(attribute_groep_id: int, aantal_voorwaarden_voor_true: int, koppelingen: List[kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKo
```

Basisklassen: `object`

Klasse voor attribuutgroep.

test(subtraject: kerapu.lbz.Subtraject.Subtraject) → `bool`

Test of een subtraject voldoet aan de voorwaarden van deze attribuutgroep.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `bool`

kerapu.boom.BeslisRegel module

```
class kerapu.boom.BeslisRegel.BeslisRegel(beslist_regel_id: int, attribuut_groep: kerapu.boom.AttribuutGroep.AttribuutGroep, label_true: str, label_false: str, indicatie_aanspraakbeperking: bool)
```

Basisklassen: `object`

Klasse voor beslisregels.

klim(subtraject: kerapu.lbz.Subtraject.Subtraject) → `str`

Klimt door de beslisboom een geeft het uiteindelijk gevonden label terug.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `str`

verrijkt(beslist_regel_true, beslist_regel_false) → `None`

Verrijkt deze beslisregel met beslisregels voor True and False.

Parameters

- `beslist_regel_true` (`kerapu.boom.BeslisRegel.BeslisRegel`) – De beslisregel voor True.

- **beslist_regel_false** (`kerapu.boom.BeslisRegel.BeslisRegel`) – De beslisregel voor False.

kerapu.boom.ZorgProductGroep module

class `kerapu.boom.ZorgProductGroep.ZorgProductGroep(zorg_product_groep_code: str)`
 Basisklassen: `object`

Klasse voor zorgproductgroepen.

klim(`subtraject: kerapu.lbz.Subtraject.Subtraject`) → `str`
 Bepaalt de zorgproductgroepcode van een subtraject.

Parameters `subtraject` (`subtraject`) – Het subtraject.

versie_toevoegen(`versie: kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie`) → `None`
 Voegt een versie toe aan de lijst met versies voor dit zorgproductgroep.

Parameters `versie` (`ZorgProductGroepVersie`) – De toe te voegen versie.

kerapu.boom.ZorgProductGroepVersie module

class `kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie(zorg_product_groep_code: str, beslis_regel: kerapu.boom.BeslisRegel.BeslisRegel, begin_datum: str, eind_datum: str)`

Basisklassen: `object`

Klasse voor versies zorgproductgroepversies.

is_actueel(`subtraject: kerapu.lbz.Subtraject.Subtraject`) → `bool`
 Geeft True als deze versie actueel is voor een subtraject, anders False.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `bool`

klim(`subtraject: kerapu.lbz.Subtraject.Subtraject`) → `str`
 Klimt door de beslisboom een geeft het uiteindelijk gevonden label terug.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `str`

Module contents

kerapu.command package

Submodules

kerapu.command.ShredderCommand module

class kerapu.command.ShredderCommand. **ShredderCommand** (*name=None*)

Basisklassen: cleo.commands.command.Command

Converteert XML-bestanden met groupertabellen naar CSV-bestanden

kerapu:shredder {XML-bestand : XML-bestand met groupertabellen, b.v. BoomBestanden.xml, Referenties.xml} {folder : Folder waar de CSV-bestanden moeten worden opgeslagen}

handle() → int

Executes the command.

kerapu.command.TestsetShredderCommand module

class kerapu.command.TestsetShredderCommand. **TestShredderCommand** (*name=None*)

Basisklassen: cleo.commands.command.Command

Converteert XML-bestand met de testset naar een CSV-bestand

kerapu:test-shredder {testset-zip : ZIP-bestand met de testset} {testset-csv : Path waar het CSV-bestand met de tests moeten worden opgeslagen}

handle() → int

Executes the command.

static ordinal(path: str) → int

Geeft het volgnummer van een test.

Parameters **path** (*str*) – Het path naar het XML-bestand met de test case.

Module contents

kerapu.lbz package

Submodules

kerapu.lbz.Diagnose module

class kerapu.lbz.Diagnose. **Diagnose** (*specialisme_code: str, diagnose_code: str*)

Basisklassen: **object**

Klasse voor diagnosen.

diagnose_attribute_aantal(diagnose_attribute_code: str, datum: str) → int

Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voldoet aan een (specialismecode, diagnosecode) op een peildatum.

Parameters

- **diagnose_attribute_code** (*str*) – De attribuutcode voor de (specialismecode, diagnosecode) combinatie.
- **datum** (*str*) – De peildatum.

Return type `int`

diagnose_cluster_aantal (*cluster_code: str, cluster_nummer: int, datum: str*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voorkomt in een diagnosecodecluster op een peildatum.

Parameters

- **cluster_code** (*str*) – De diagnoseclustercode.
- **cluster_nummer** (*int*) – De clusternummer (0..6).
- **datum** (*str*) – De peildatum.

Return type `int`

static init_static (*folder: str*) → `None`

Initialiseert alle statistische data.

Parameters `folder` (*str*) – De folder met alle goupertabellen.

kerapu.lbz.Patient module

class `kerapu.lbz.Patient`.**Patient** (*geboorte_datum: str, geslacht_code: str*)

Basisklassen: `object`

Klasse voor patiëntenn.

property geslacht_code

Geeft het geslacht van deze patiënt.

Return type `str`

leeftijd (*datum: str*) → `int`

Geeft de leeftijd van deze patient op een peildatum.

Parameters `datum` (*str*) – De peildatum.

Return type `int`

static normaliseer_geslacht_code (*geslacht_code: str*) → `str`

Normaliseert een geslachtscode naar 1 (man), 2 (vrouw) of 9 (anders).

Parameters `geslacht_code` (*str*) – De geslachtscode.

Return type `str`

kerapu.lbz.Specialisme module

```
class kerapu.lbz.Specialisme.Specialisme(specialisme_code: str)
    Basisklassen: object
```

Klasse voor specialismen.

```
static init_static(folder: str) → None
    Initialiseert alle statistische data.
```

Parameters `folder (str)` – De folder met alle goupertabellen.

```
specialisme_aantal(specialisme_code: str, datum: str) → int
```

Geeft het aantal malen (d.w.z. 0 of 1) dat dit specialisme voldoet aan een attributecode op een gegeven datum . :param str specialisme_code: De attribuutcode waaraan voldaan moet worden. :param str datum: De datum.

Return type `int`

```
specialisme_cluster_aantal(cluster_code: str, cluster_nummer: int, datum: str) → int
```

Geeft het aantal malen (d.w.z. 0 of 1) dat dit specialisme voldoet aan een clustercode op een gegeven datum.

Parameters

- `cluster_code (str)` – De clustercode waaraan voldaan moet worden.
- `cluster_nummer (int)` – Het clusternummer.
- `datum (str)` – De datum.

Return type `int`

kerapu.lbz.Subraject module

```
class kerapu.lbz.Subraject.Subraject(subraject_nummer: str, specialisme_code: str, diagnose_code: str, zorg_type_code: str, zorg_vraag_code: str, begin_datum: str, geboorte_datum: str, geslacht_code: str, zorg_instelling_code: str)
```

Basisklassen: `object`

Klasse voor subrajecten.

```
add_zorg_activiteit(zorg_activiteit_code: str, aantal: int) → None
    Voegt een zorgactiviteit toe and dit subraject.
```

Parameters

- `zorg_activiteit_code (str)` – De zorgactiviteitcode.
- `aantal (int)` – Het aantal malen (of eenheden) dat de zorgactiviteit is uitgevoerd.

```
property begin_datum
```

Geeft de begindatum van dit subraject.

Return type `str`

```
property leeftijd
```

Geeft de leeftijd van de patient van dit subraject.

Return type `int`

property subtraject_nummer
Geeft het subtrajectnummer van dit subtraject.

Return type str

telling_behandel_klasse (behandel_klasse_code: str, weeg_factor_nummer: int) → int
Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voorkomt in een behandelklasse.

Parameters

- **behandel_klasse_code** (str) – De behandelklassecode waartegen getest moet worden.
- **weeg_factor_nummer** (int) – Het weegfactornummer (0..2).

Return type int

telling_diagnose_attribuut (diagnose_attribuut_code: str) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat de diagnose van dit subtraject voldoet aan een (specialismecode, diagnosecode) combinatie.

Parameters **diagnose_attribuut_code** (str) – De attribuutcode voor de (specialisme-code, diagnosecode) combinatie.

Return type int

telling_diagnose_cluster (cluster_code: str, cluster_nummer: int) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat in dit subtraject voldoet aan een diagnoseclustercode.

Parameters

- **cluster_code** (str) – De cluster_code waartegen getest moet worden.
- **cluster_nummer** (int) – Het clusternummer (1..6).

Return type int

telling_geslacht_code (geslacht_code: str) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat de patient van dit subtraject voldoet aan een geslacht.

Parameters **geslacht_code** (str) – De geslachtscode waartegen getest moet worden.

Return type int

telling_specialisme (specialisme_code: str) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van dit subtraject voldoet aan een specialismecode.

Parameters **specialisme_code** (str) – De specialismecode.

Return type int

telling_specialisme_cluster (cluster_code: str, cluster_nummer: int) → int
Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van dit subtraject voldoet aan een specialismecluster.

Parameters

- **cluster_code** (str) – De clustercode waartegen getest moet worden.
- **cluster_nummer** (int) – Het clusternummer (1..2).

Return type int

telling_zorg_activiteit (*zorg_activiteit_code: str, weeg_factor_nummer: int*) → *int*

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).

Return type *int*

telling_zorg_activiteit_cluster (*cluster_code: str, cluster_nummer: int, weeg_factor_nummer: int*) → *int*

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **cluster_nummer** (*int*) – Het clusternummer (1..10).
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).

Return type *int*

telling_zorg_instelling (*agb_code: str*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorginstelling van dit subtraject voldoet aan AGB-code.

Parameters **agb_code** (*str*) – De AGB-code waaraan de zorginstelling moet voldoen.

Return type *int*

telling_zorg_type_attribuut (*zorg_type_attribuut_code: str*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgtype van dit subtraject voldoet aan een (specialismecode, zorgtypecode) combinatie.

Parameters **zorg_type_attribuut_code** (*str*) – De attribuutcode voor de (specialismecode, zorgtypecode) combinatie.

Return type *int*

telling_zorg_vraag_attribuut (*zorg_vraag_attribuut_code: str*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van dit subtraject voldoet aan een (specialismecode, zorgvraagcode) combinatie.

Parameters **zorg_vraag_attribuut_code** (*str*) – De attribuutcode voor de (specialismecode, zorgvraagcode) combinatie.

Return type *int*

telling_zorg_vraag_cluster (*cluster_code: str, cluster_nummer: int*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voorkomt in een zorgvraagcluster.

Parameters

- **cluster_code** (*str*) – De cluster_code waartegen getest moet worden.
- **cluster_nummer** (*int*) – Het clusternummer (1..2).

Return type *int*

property zorg_product_code

Geeft de zorgproductcode van dit subtraject.

Return type *str|None*

property zorg_product_groep_code
Geeft de zorgproductgroepcode van dit subtraject.

Return type str|None

kerapu.lbz.ZorgActiviteit module

class kerapu.lbz.ZorgActiviteit.**ZorgActiviteit** (*zorg_activiteit_code: str, aantal: int*)
Basisklassen: `object`

Klasse voor zorgactiviteiten.

behandel_klasse_aantal (*zorg_product_groep_code: str, behandel_klasse_code: str, weeg_factor_nummer: int, datum: str*) → int

Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voorkomt in een behandeklasse op een peildatum.

Parameters

- **zorg_product_groep_code** (*str*) – De zorgproductgroepcode van het subtraject van deze zorgactiviteit.
- **behandel_klasse_code** (*str*) – De behandelklasse.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).
- **datum** (*str*) – De peildatum.

Return type int

static init_static (*folder: str*) → None
Initialiseert alle statistische data.

Parameters **folder** (*str*) – De folder met alle goupertabellen.

zorg_activiteit_aantal (*zorg_activiteit_code: str, weeg_factor_nummer: int, datum: str*) → int
Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).
- **datum** (*str*) – De peildatum.

Return type int

zorg_activiteit_cluster_aantal (*cluster_code: str, cluster_nummer: int, weeg_factor_nummer: int, datum: str*) → int
Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **cluster_nummer** (*int*) – het cluster nummber (1..10).
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).
- **datum** (*str*) – De peildatum.

Return type int

kerapu.lbz.ZorgInstelling module

```
class kerapu.lbz.ZorgInstelling.ZorgInstelling(zorg_instelling_code: str)
    Basisklassen: object
```

Klasse voor zorginstellingen.

```
zorg_instelling_aantal(zorg_instelling_code: str) → int
```

Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorginstelling voldoet aan een zorginstellingcode.

Parameters `zorg_instelling_code (str)` – De AGB-code van de zorginstelling.

Return type `int`

kerapu.lbz.ZorgType module

```
class kerapu.lbz.ZorgType.ZorgType(specialisme_code: str, zorg_type_code: str)
    Basisklassen: object
```

Klasse voor zorgtypen.

```
static init_static(folder: str) → None
```

Initialiseert alle statistische data.

Parameters `folder (str)` – De folder met alle goupertabellen.

```
zorg_type_attribute_aantal(zorg_type_attribute_code: str, datum: str) → int
```

Geeft het aantal malen (d.w.z. 0 of 1) dat deze diagnose voldoet aan een (specialismecode, zorgtypecode) combinatie op een peildatum.

Parameters

- `zorg_type_attribute_code (str)` – De attribuutcode voor (specialismecode, diagnosticode) combinatie.
- `datum (str)` – De peildatum.

Return type `int`

```
zorg_type_cluster_aantal(cluster_code: str, cluster_nummer: int, datum: str) → int
```

Geeft het aantal malen (d.w.z. 0 of 1) dat deze zorgtype voorkomt in een zorgtypecluster op een peildatum.

Parameters

- `cluster_code (str)` – De zorgtypeclustercode.
- `cluster_nummer (int)` – Het clusternummer (0..2).
- `datum (str)` – De peildatum.

Return type `int`

kerapu.lbz.ZorgVraag module

```
class kerapu.lbz.ZorgVraag.ZorgVraag(specialisme_code: str, zorg_vraag_code: str)  
    Basisklassen: object
```

VII

ପ୍ରକାଶକ ପତ୍ର

static init static [vader: str] / None
Initialiseert alle statistische data

Initiativsekt an die statistische data.

Parameters `folder` (*str*) – De folder met alle goupertabellen.

zorg_vraag_attribute_aantal(*zorg_vraag_attribuut_code*: str, *datum*: str) → int

Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voldoet aan een (specialismecode, zorgvraagcode) combinatie op een peildatum.

Parameters

- **zorg_vraag_attribuut_code** (*str*) – De attribuutcode voor (specialismecode, diagnosecode) combinatie.
 - **datum** (*str*) – De peildatum.

Return type int

zorg_vraag_cluster_aantal(cluster_code: str, cluster_nummer: int, datum: str) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat deze zorgvraag voorkomt in een zorgvraagcluster op een peildatum.

Parameters

- **cluster_code** (*str*) – De zorgvraagclustercode.
 - **cluster_nummer** (*int*) – Het clusternummer (0..2).
 - **datum** (*str*) – De peildatum.

Return type int

Module contents

kerapu.shredder package

Submodules

kerapu.shredder.BoomBestandenShredder module

```
class kerapu.shredder.BoomBestandenShredder BoomBestandenShredder(io::
```

```
kerapu.style  
target_dir:  
str)
```

Basisklassen: *kerapu.shredder.Shredder.Shredder*

Klasse voor het schreden en opslaan in CSV-formaat van boombestanden opgeslagen in XML-formaat.

shred_xml_file(*filename*: *str*) → *None*

Slaat de boombestanden op in CSV-formaat.

Parameters `filename` (*str*) – De filenaam van het XML bestand.

kerapu.shredder.ReferentieShredder module

```
class kerapu.shredder.ReferentieShredder(io:  
                                         kerapu.style.KerapuStyle.KerapuStyle,  
                                         target_dir: str)
```

Basisklassen: `kerapu.shredder.Shredder.Shredder`

Klasse voor het schreden en opslaan in CSV-formaat van referentietabellen opgeslagen in XML-formaat.

`shred_xml_file(filename: str) → None`

Slaat de referentietabellen op in CSV-formaat.

Parameters `filename (str)` – De filenaam van het XML bestand.

kerapu.shredder.Shredder module

```
class kerapu.shredder.Shredder(io: kerapu.style.KerapuStyle.KerapuStyle, tar-  
                                get_dir: str)
```

Basisklassen: `object`

Klasse voor het schreden van XML-bestanden en opslaan in CSV-formaat.

`static extract_field(element: lxml.etree.Element, tag: str) → str`

Extracts de waarde van een XML element.

Parameters

- `element (Element)` – Het parent XML element.
- `tag (str)` – De tag van het gevraagde XML-element.

Return type `str`

`extract_table(table: lxml.etree.Element, filename: str, fields: List, xpaths: List) → None`

Extracts een groupertabel uit XML een slaat de tabel op in een CSV-bestand.

Parameters

- `table (Element)` – De naam van de groupertabel.
- `filename (str)` – De filenaam van het CSV-bestand.
- `fields (list)` – Een lijst met velden (d.w.z. kolommen in het CSV-bestand).
- `xpaths (list)` – Een lijst met xpath voor het extracten van de bovenstaande velden.

Module contents

kerapu.style package

Submodules

kerapu.style.KerapuStyle module

```
class kerapu.style.KerapuStyle(input_object: cleo.inputs.input.Input, out-  
                               put_object: cleo.outputs.output.Output)
```

Basisklassen: `cleo.styles.cleo_style.CleoStyle`

Output style for Kerapu.

text (*message*: *Optional[Union[str, list]]*) → *None*
Formats informational text.

Module contents

5.1.2 Submodules

5.1.3 kerapu.Kerapu module

class `kerapu.Kerapu`

Basisklassen: `object`

Een implementatie van de grouper in Python.

bepaal_zorg_product (*subtraject*: `kerapu.lbz.Subtraject.Subtraject`) → `str`

Bepaalt de zorgproduct van een subtraject.

Parameters `subtraject` (`Subtraject`) – Het subtraject waarvoor de zorgproductcode moet worden bepaalt.

Return type `str`

bepaal_zorg_product_groep (*subtraject*: `kerapu.lbz.Subtraject.Subtraject`) → `str`

Bepaalt de zorgproductgroep van een subtraject.

Parameters `subtraject` (`Subtraject`) – Het subtraject waarvoor de zorgproductgroep moet worden bepaalt.

Return type `str`

init_static (*folder*: `str`) → *None*

Initialiseert alle statistische data.

Parameters `folder` (`str`) – De folder met alle goupertabellen.

5.1.4 Module contents

`kerapu.clean_bool` (*x*: `str`) → `bool`

`kerapu.clean_code` (*code*: `str`, *lengte*: `int`) → `str`

Schoont een code van voor- en naloop whitespace en voorziet de code van het juiste aantal voorloop nullen.

Parameters

- `code` (`str`) – De code.
- `lengte` (`int`) – De gewenste lengte van de code.

Return type `str`

`kerapu.clean_date` (*x*: `str`) → `str`

`kerapu.clean_int` (*x*: `Any`, *leeg*: *Optional[int]* = `None`) → *Optional[int]*

`kerapu.clean_str` (*x*: `Any`) → *Optional[str]*

Python-moduleindex

k

kerapu, 31
kerapu.application, 12
kerapu.application.kerapu, 11
kerapu.application.KerapuApplication,
 11
kerapu.boom, 22
kerapu.boom.attribuut, 13
kerapu.boom.attribuut.Attribuut, 12
kerapu.boom.attribuut.Attribuut12, 12
kerapu.boom.attribuut.Attribuut21, 12
kerapu.boom.attribuut_groep_koppeling,
 14
kerapu.boom.attribuut_groep_koppeling.ATTRIBUUT_KERAPU, 31
kerapu.boom.attribuut_groep_koppeling.ATTRIBUUT_KERAPU_KOPPELING,
 13
kerapu.boom.attribuut_groep_koppeling.ATTRIBUUT_KERAPU_KOPPELING,
 14
kerapu.boom.ATTRIBUUTGroep, 20
kerapu.boom.BeslisRegel, 20
kerapu.boom.boom_parameter, 20
kerapu.boom.boom_parameter.BehandelKlasse,
 15
kerapu.boom.boom_parameter.BoomParameter,
 15
kerapu.boom.boom_parameter.DiagnoseCluster,
 15
kerapu.boom.boom_parameter.DiagnoseCode,
 16
kerapu.boom.boom_parameter.Geslacht, 16
kerapu.boom.boom_parameter.Leeftijd, 16
kerapu.boom.boom_parameter.SpecialismeCluster,
 17
kerapu.boom.boom_parameter.SpecialismeCode,
 17
kerapu.boom.boom_parameter.ZorgActiviteitCluster,
 17
kerapu.boom.boom_parameter.ZorgActiviteitCode,
 18
kerapu.boom.boom_parameter.ZorgInstellingCode,
 18
kerapu.boom.boom_parameter.ZorgTypeCode,
 19
kerapu.boom.boom_parameter.ZorgVraagCluster,
 19
kerapu.boom.boom_parameter.ZorgVraagCode,
 19
kerapu.boom.ZorgProductGroep, 21
kerapu.boom.ZorgProductGroepVersie, 21
kerapu.command, 22
kerapu.command.ShredderCommand, 22
kerapu.command.TestsetShredderCommand,
 22
kerapu.lbz, 29
kerapu.lbz.ATTRIBUUT_GROEP_DIAGNOSE, 22
kerapu.lbz.Patiënt, 23
kerapu.lbz.Specialisme, 24
kerapu.lbz.Subtraject, 24
kerapu.lbz.ZorgActiviteit, 27
kerapu.lbz.ZorgInstelling, 28
kerapu.lbz.ZorgType, 28
kerapu.lbz.ZorgVraag, 29
kerapu.shredder, 30
kerapu.shredder.BoomBestandenShredder,
 29
kerapu.shredder.ReferentieShredder, 30
kerapu.shredder.Shredder, 30
kerapu.style, 31
kerapu.style.KerapuStyle, 30

Index

A

add_zorg_activiteit()
 (*kerapu.lbz.Subraject.Subraject methode*), 24
Attribuut (*klasse in kerapu.boom.attribuut.Attribuut*),
 12
Attribuut12 (*klasse*
 kerapu.boom.attribuut.Attribuut12), 12
Attribuut21 (*klasse*
 kerapu.boom.attribuut.Attribuut21), 12
AttribuutGroep (*klasse*
 kerapu.boom.AttribuutGroep), 20
AttribuutGroepKoppeling (*klasse* *in*
 kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling),
 13
AttribuutGroepKoppeling2 (*klasse* *in*
 kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2),
 14

B

begin_datum() (*kerapu.lbz.Subraject.Subraject*
 property), 24
behandel_klasse_aantal()
 (*kerapu.lbz.ZorgActiviteit.ZorgActiviteit*
 methode), 27
BehandelKlasse (*klasse* *in*
 kerapu.boom.boom_parameter.BehandelKlasse),
 15
bepaal_zorg_product() (*kerapu.Kerapu.Kerapu*
 methode), 31
bepaal_zorg_product_groep()
 (*kerapu.Kerapu.Kerapu* *methode*), 31
BeslisRegel (*klasse in kerapu.boom.BeslisRegel*), 20
BoomBestandenShredder (*klasse* *in*
 kerapu.shredder.BoomBestandenShredder),
 29
BoomParameter (*klasse* *in*
 kerapu.boom.boom_parameter.BoomParameter),
 15

C

clean_bool () (*in module kerapu*), 31
clean_code () (*in module kerapu*), 31
clean_date () (*in module kerapu*), 31
clean_int () (*in module kerapu*), 31
clean_str () (*in module kerapu*), 31
create_boom_parameter () (*in* *module*
 kerapu.boom.boom_parameter), 20
in D

Diagnose (*klasse in kerapu.lbz.Diagnose*), 22
 in diagnose_attribute_aantal()
 (*kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling*),
 13
 in diagnose_cluster_aantal()
 (*kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2*),
 14
DiagnoseDiagnoseCluster (*klasse* *in*
 kerapu.boom.boom_parameter.DiagnoseCluster),
 15

DiagnoseDiagnoseCode (*klasse* *in*
 kerapu.boom.boom_parameter.DiagnoseCode),
 16

E

extract_field() (*statische* *methode* *van*
 kerapu.shredder.Shredder.Shredder), 30
extract_table() (*kerapu.shredder.Shredder.Shredder*
 methode), 30

G

Geslacht (*klasse* *in*
 kerapu.boom.boom_parameter.Geslacht),
 16
geslacht_code () (*kerapu.lbz.Patient.Patient* *prop-*
 erty), 23
get_default_commands ()
 (*kerapu.application.KerapuApplication.KerapuApplication*
 methode), 11

H

handle() (*kerapu.command.ShredderCommand.ShredderCommand methode*), 22
handle() (*kerapu.command.TestsetShredderCommand.TestShredderCommand methode*), 22

I

init_static() (*kerapu.Kerapu.Kerapu methode*), 31
init_static() (*statische methode van kerapu.lbz.Diagnose.Diagnose*), 23
init_static() (*statische methode van kerapu.lbz.Specialisme.Specialisme*), 24
init_static() (*statische methode van kerapu.lbz.ZorgActiviteit.ZorgActiviteit*), 27
init_static() (*statische methode van kerapu.lbz.ZorgType.ZorgType*), 28
init_static() (*statische methode van kerapu.lbz.ZorgVraag.ZorgVraag*), 29
is_actueel() (*kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie methode*), 21

K

kerapu
 module, 31
Kerapu (*klasse in kerapu.Kerapu*), 31
kerapu.application
 module, 12
kerapu.application.kerapu
 module, 11
kerapu.application.KerapuApplication
 module, 11
kerapu.boom
 module, 22
kerapu.boom.attribuut
 module, 13
kerapu.boom.attribuut.Attribuut
 module, 12
kerapu.boom.attribuut.Attribuut12
 module, 12
kerapu.boom.attribuut.Attribuut21
 module, 12
kerapu.boom.attribuut_groep_koppeling
 module, 14
kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling
 module, 13
kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling
 module, 14
kerapu.boom.AttribuutGroep
 module, 20
kerapu.boom.BeslisRegel
 module, 20
kerapu.boom.boom_parameter
 module, 20

 kerapu.boom.boom_parameter.BehandelKlasse
 module, 15
 kerapu.boom.boom_parameter.BoomParameter
 module, 15
 kerapu.boom.boom_parameter.DiagnoseCluster
 module, 15
 kerapu.boom.boom_parameter.DiagnoseCode
 module, 16
 kerapu.boom.boom_parameter.Geslacht
 module, 16
 kerapu.boom.boom_parameter.Leeftijd
 module, 16
 kerapu.boom.boom_parameter.SpecialismeCluster
 module, 17
 kerapu.boom.boom_parameter.SpecialismeCode
 module, 17
 kerapu.boom.boom_parameter.ZorgActiviteitCluster
 module, 17
 kerapu.boom.boom_parameter.ZorgActiviteitCode
 module, 18
 kerapu.boom.boom_parameter.ZorgInstellingCode
 module, 18
 kerapu.boom.boom_parameter.ZorgTypeCode
 module, 19
 kerapu.boom.boom_parameter.ZorgVraagCluster
 module, 19
 kerapu.boom.boom_parameter.ZorgVraagCode
 module, 19
 kerapu.boom.ZorgProductGroep
 module, 21
 kerapu.boom.ZorgProductGroepVersie
 module, 21
 kerapu.command
 module, 22
 kerapu.command.ShredderCommand
 module, 22
 kerapu.command.TestsetShredderCommand
 module, 22
 kerapu.Kerapu
 module, 31
 kerapu.lbz
 module, 29
 kerapu.lbz.Diagnose
 module, 22
 kerapu.lbz.Patient
 module, 23
 kerapu.lbz.Specialisme
 module, 24
 kerapu.lbz.Subraject
 module, 24
 kerapu.lbz.ZorgActiviteit
 module, 27
 kerapu.lbz.ZorgInstelling
 module, 28

kerapu.lbz.ZorgType
 module, 28
 kerapu.lbz.ZorgVraag
 module, 29
 kerapu.shredder
 module, 30
 kerapu.shredder.BoomBestandenShredder
 module, 29
 kerapu.shredder.RiferentieShredder
 module, 30
 kerapu.shredder.Shredder
 module, 30
 kerapu.style
 module, 31
 kerapu.style.KerapuStyle
 module, 30
 KerapuApplication (klasse in
 kerapu.application.KerapuApplication), 11
 KerapuStyle (klasse in kerapu.style.KerapuStyle), 30
 klim() (kerapu.boom.BeslisRegel.BeslisRegel me-
 thode), 20
 klim() (kerapu.boom.ZorgProductGroep.ZorgProductGroep
 methode), 21
 klim() (kerapu.boom.ZorgProductGroepVersie.ZorgProductGroep
 methode), 21

L

Leeftijd (klasse in
 kerapu.boom.boom_parameter.Leeftijd),
 16
 leeftijd() (kerapu.lbz.Patient.Patient methode), 23
 leeftijd() (kerapu.lbz.Subraject.Subraject
 property), 24

M

maak_attribuut() (in module
 kerapu.boom.attribuut), 13
 maak_attribuut_groep_koppeling() (in mo-
 dule kerapu.boom.attribuut_groep_koppeling),
 14
 main() (in module kerapu.application.kerapu), 11
 module
 kerapu, 31
 kerapu.application, 12
 kerapu.application.kerapu, 11
 kerapu.application.KerapuApplication,
 11
 kerapu.boom, 22
 kerapu.boom.attribuut, 13
 kerapu.boom.attribuut.Attribuut, 12
 kerapu.boom.attribuut.Attribuut12,
 12
 kerapu.boom.attribuut.Attribuut21,
 12

kerapu.boom.attribuut_groep_koppeling,
 14
 kerapu.boom.attribuut_groep_koppeling.Attribuut
 13
 kerapu.boom.attribuut_groep_koppeling.Attribuut
 14
 kerapu.boom.AttribuutGroep, 20
 kerapu.boom.BeslisRegel, 20
 kerapu.boom.boom_parameter, 20
 kerapu.boom.boom_parameter.BehandelKlasse,
 15
 kerapu.boom.boom_parameter.BoomParameter,
 15
 kerapu.boom.boom_parameter.DiagnoseCluster,
 15
 kerapu.boom.boom_parameter.DiagnoseCode,
 16
 kerapu.boom.boom_parameter.Geslacht,
 16
 kerapu.boom.boom_parameter.Leeftijd,
 16
 kerapu.boom.boom_parameter.SpecialismeCluster,
 17
 kerapu.boom.boom_parameter.SpecialismeCode,
 17
 kerapu.boom.boom_parameter.ZorgActiviteitCluste
 17
 kerapu.boom.boom_parameter.ZorgActiviteitCode,
 18
 kerapu.boom.boom_parameter.ZorgInstellingCode,
 18
 kerapu.boom.boom_parameter.ZorgTypeCode,
 19
 kerapu.boom.boom_parameter.ZorgVraagCluster,
 19
 kerapu.boom.boom_parameter.ZorgVraagCode,
 19
 kerapu.boom.ZorgProductGroep, 21
 kerapu.boom.ZorgProductGroepVersie,
 21
 kerapu.command, 22
 kerapu.command.ShredderCommand, 22
 kerapu.command.TestsetShredderCommand,
 22
 kerapu.Kerapu, 31
 kerapu.lbz, 29
 kerapu.lbz.Diagnose, 22
 kerapu.lbz.Patient, 23
 kerapu.lbz.Specialisme, 24
 kerapu.lbz.Subraject, 24
 kerapu.lbz.ZorgActiviteit, 27
 kerapu.lbz.ZorgInstelling, 28
 kerapu.lbz.ZorgType, 28
 kerapu.lbz.ZorgVraag, 29

kerapu.shredder, 30
kerapu.shredder.BoomBestandenShredder, 29
kerapu.shredder.RiferentieShredder, 30
kerapu.shredder.Shredder, 30
kerapu.style, 31
kerapu.style.KerapuStyle, 30

N

normaliseer_geslacht_code() (statische methode van kerapu.lbz.Patient.Patient), 23

O

ordinal() (statische methode van kerapu.command.TestsetShredderCommand.TestShredderCommand), 22

P

Patient (klasse in kerapu.lbz.Patient), 23

R

ReferentieShredder (klasse kerapu.shredder.ReferentieShredder), 30

S

shred_xml_file() (kerapu.shredder.BoomBestandenShredder.BoomBestandenShredder methode), 29
shred_xml_file() (kerapu.shredder.ReferentieShredder.ReferentieShredder methode), 30
Shredder (klasse in kerapu.shredder.Shredder), 30
ShredderCommand (klasse kerapu.command.ShredderCommand), 22
Specialisme (klasse in kerapu.lbz.Specialisme), 24
specialisme_aantal()
 (kerapu.lbz.Specialisme.Specialisme methode), 24
specialisme_cluster_aantal()
 (kerapu.lbz.Specialisme.Specialisme methode), 24
SpecialismeCluster (klasse kerapu.boom.boom_parameter.SpecialismeCluster) in
 (kerapu.boom.boom_parameter.SpecialismeCluster) 17
SpecialismeCode (klasse kerapu.boom.boom_parameter.SpecialismeCode) in
 (kerapu.boom.boom_parameter.SpecialismeCode), 17
Subraject (klasse in kerapu.lbz.Subraject), 24
subraject_nummer()
 (kerapu.lbz.Subraject.Subraject property), 24

T

tel() (kerapu.boom.attribuut.AttribuutAttribuut methode), 12
tel() (kerapu.boom.attribuut.AttribuutAttribuut methode), 12
tel() (kerapu.boom.boom_parameter.BehandelKlasse.BehandelKlasse methode), 15
tel() (kerapu.boom.boom_parameter.BoomParameter.BoomParameter methode), 15
tel() (kerapu.boom.boom_parameter.DiagnoseCluster.DiagnoseCluster methode), 15
tel() (kerapu.boom.boom_parameter.DiagnoseCode.DiagnoseCode methode), 16
tel() (kerapu.boom.boom_parameter.Geslacht.Geslacht methode), 16
tel() (kerapu.boom.boom_parameter.Leeftijd.Leeftijd methode), 16
tel() (kerapu.boom.boom_parameter.SpecialismeCluster.SpecialismeCluster methode), 17
tel() (kerapu.boom.boom_parameter.SpecialismeCode.SpecialismeCode methode), 17
tel() (kerapu.boom.boom_parameter.ZorgActiviteitCluster.ZorgActiviteitCluster methode), 17
tel() (kerapu.boom.boom_parameter.ZorgActiviteitCode.ZorgActiviteitCode methode), 18
tel() (kerapu.boom.boom_parameter.ZorgInstellingCode.ZorgInstellingCode methode), 18
tel() (kerapu.boom.boom_parameter.ZorgTypeCode.ZorgTypeCode methode), 19
tel() (kerapu.boom.boom_parameter.ZorgVraagCluster.ZorgVraagCluster methode), 19
tel() (kerapu.boom.boom_parameter.ZorgVraagCode.ZorgVraagCode methode), 19
telling_behandel_klasse()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_diagnose_attribuut()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_diagnose_cluster()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_geslacht_code()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_specialisme()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_zorg_activiteit()
 (kerapu.lbz.Subraject.Subraject methode), 25
telling_zorg_activiteit_cluster()
 (kerapu.lbz.Subraject.Subraject methode), 26
telling_zorg_instelling()
 (kerapu.lbz.Subraject.Subraject methode), 26
telling_zorg_type_attribuut()
 (kerapu.lbz.Subraject.Subraject methode), 26
telling_zorg_vraag_attribuut()
 (kerapu.lbz.Subraject.Subraject methode), 26

		18		
telling_zorg_vraag_cluster()	(kerapu.lbz.Subtraject.Subtraject methode), 26	ZorgInstelling	(klasse	in
test ()	(kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling.ZorgInstellingKoppeling methode), 13	ZorgInstellingCode	(klasse	in
test ()	(kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling.ZorgInstellingKoppeling methode), 14	18		
test ()	(kerapu.boom.AttribuutGroep.AttribuutGroep methode), 20	ZorgProductGroep	(klasse	in
		kerapu.boom.ZorgProductGroep), 21		
TestShredderCommand	(klasse in kerapu.command.TestsetShredderCommand), 22	ZorgProductGroepVersie	(klasse	in
		kerapu.boom.ZorgProductGroepVersie), 21		
text ()	(kerapu.style.KerapuStyle.KerapuStyle methode), 30	ZorgType	(klasse in kerapu.lbz.ZorgType), 28	
		ZorgTypeCode	(klasse	in
		kerapu.boom.boom_parameter.ZorgTypeCode), 19		
V		ZorgVraag	(klasse in kerapu.lbz.ZorgVraag), 29	
verrijk ()	(kerapu.boom.BeslisRegel.BeslisRegel methode), 20	ZorgVraagCluster	(klasse	in
versie_toevoegen ()	(kerapu.boom.ZorgProductGroep.ZorgProductGroep methode), 21	kerapu.boom.boom_parameter.ZorgVraagCluster), 19		
Z		ZorgVraagCode	(klasse	in
zorg_activiteit_aantal ()	(kerapu.lbz.ZorgActiviteit.ZorgActiviteit methode), 27	kerapu.boom.boom_parameter.ZorgVraagCode), 19		
zorg_activiteit_cluster_aantal ()	(kerapu.lbz.ZorgActiviteit.ZorgActiviteit methode), 27			
zorg_instelling_aantal ()	(kerapu.lbz.ZorgInstelling.ZorgInstelling methode), 28			
zorg_product_code ()	(kerapu.lbz.Subtraject.Subtraject property), 26			
zorg_product_groep_code ()	(kerapu.lbz.Subtraject.Subtraject property), 26			
zorg_type_attribute_aantal ()	(kerapu.lbz.ZorgType.ZorgType methode), 28			
zorg_type_cluster_aantal ()	(kerapu.lbz.ZorgType.ZorgType methode), 28			
zorg_vraag_attribute_aantal ()	(kerapu.lbz.ZorgVraag.ZorgVraag methode), 29			
zorg_vraag_cluster_aantal ()	(kerapu.lbz.ZorgVraag.ZorgVraag methode), 29			
ZorgActiviteit	(klasse in kerapu.lbz.ZorgActiviteit), 27			
ZorgActiviteitCluster	(klasse in kerapu.boom.boom_parameter.ZorgActiviteitCluster), 17			
ZorgActiviteitCode	(klasse in kerapu.boom.boom_parameter.ZorgActiviteitCode),			