

Dokumentation

RestAPI

PersonAPI - PersonNotFoundException

POST

api/person/	Creating a new person
--------------------	-----------------------

JSON

@Consumes

```
{"firstName" : "?", "lastName" : "?". "email" : "?"}
```

@Produces

```
{"firstName" : "?", "lastName" : "?". "email" : "?"}
```

GET

api/person/{id}	Return a person given by the Id with all infos
api/person/phone/{number}	Return a person given by the phone number with all infos
api/person/hobby/{hobby}	Return a person given by the hobby with all infos
api/person/hobby/count/{hobby}	Return multiple persons with the given hobby with all infos
api/person/city/{zipcode}	Return multiple persons which lives in the given city with all infos
api/person/complete	Return all persons with all infos

api/person/{id}

JSON

@PathParam

```
{"id"}
```

@Produces

```
{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] }
```

api/person/phone/{number}

JSON

@PathParam

{"onenumber"}

@Produces

{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] }

api/person/hobby/{hobby}

JSON

@PathParam

{"hobby"}

@Produces

{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] }

api/person/hobby/count/{hobby}

;ndre til

gets the number of persons interested in a specifik hobby

JSON

@PathParam

{"hobby"}

@Produces

{ "count" : ? }

api/person/city/{zipcode}

JSON

@PathParam

{"zipcode"}

@Produces

[{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] },

{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] }, ...]

api/person/complete

JSON

@Produces

```
[{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] },  
{"id" : ?, "firstName" : "?", "lastName": "?", "email", "hobbyList": [] }, ...]
```

PUT

api/person/{id}

Updating a person

JSON

@PathParam

{"id"}

@Consumes

{"firstName" : "?", "lastName" : "?". "email" : "?"}

@Produces

{"firstName" : "?", "lastName" : "?". "email" : "?"}

DELETE

api/person/{id}

Deleting the person with the given id

api/person/id

JSON

@PathParam

{"id"}

@Produces

{"firstName" : "?", "lastName" : "?". "email" : "?"}

CompanyAPI – throws CompanyNotFoundException

POST

api/company/

Creating a new company

api/company/

JSON

@Consumes

```
{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}
```

@Produces

```
{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}
```

GET

api/company/cvr/{number}

Return a company given by cvr number with all infos

api/company/id

Return a company given by the id with all infos

api/company/phone/number

Return a company given by phone number with all infos

api/company/employees/number

Return multiple companies with employee value greater than given number

api/company/cvr/{number}

JSON

@PathParam

```
{"number"}
```

@Produces

```
{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}
```

api/company/{id}

JSON

@PathParam

{"id"}

@Produces

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}

api/company/phone/{number}

JSON

@PathParam

{"number"}

@Produces

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}

api/company/employees/{number}

JSON

@PathParam

{"number"}

@Produces

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : (> number), "marketValue": ?, "email" : "?"}

PUT

api/company

Updating a company

JSON

@PathParam

{"id"}

@Consumes

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}

@Produces

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}

DELETE

api/company/{id}

Deleting the company with the given id

api/company/id

JSON

@PathParam

{"id"}

@Produces

{"name" : "?", "description" : "?". "cvr" : "?", "numberEmployees" : ?, "marketValue": ?, "email" : "?"}

CityinfoAPI

POST

api/cityinfo/

Creating cityinfo

api/cityinfo/

JSON

@Consumes

```
{"zipCode" : "?", "city" : "?"}
```

@Produces

```
{"zipCode" : "?", "city" : "?"}
```

GET

api/cityinfo/all

Return all cityinfo with all datas

api/cityinfo/zipcode

Return cityinfo with given zipcode with all datas

api/cityinfo/all

JSON

@Produces

```
[{"zipCode" : "?", "city" : "?"}, {"zipCode" : "?", "city" : "?"}, {"zipCode" : "?", "city" : "?"}...]
```

api/cityinfo/zipcode

JSON

@PathParam

```
{"zipcode"}
```

@Produces

```
[{"zipCode" : "?", "city" : "?"}]
```

PUT

api/cityinfo

Updating cityinfo

api/cityinfo/{id}

tilf'j id

JSON

@PathParam

{"id"}

@Consumes

{"zipCode" : "?", "city" : "?"}

@Produces

{"zipCode" : "?", "city" : "?"}

DELETE

api/cityinfo/zipcode

Deleting cityinfo with the given zipcode

api/zipcode/id

JSON

@PathParam

{"id"}

@Produces

{"zipCode" : "?", "city" : "?"}

AddressAPI

POST

api/address

Creating a new address

api/cityinfo/

JSON

@Consumes

```
{"street": "?", "additionalInfo": "?", "cityInfo": "?"}
```

@Produces

```
{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } }
```

GET

api/address/getbystreet/street

Return multiple addresses with given street with all infos

api/address/all

Return all addresses with all infos

api/address/getbystreet/{street}

JSON

@PathParam

```
{"street"}
```

@Produces

```
{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } }
```

api/address/all

JSON

@Produces

```
[{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } },
```

```
{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } }, ...]
```

PUT

api/address

Updating an address

api/address/{id}

tilf'j id

JSON

@PathParam

{"id"}

@Consumes

{"street" : "?", "additionalInfo" : "?", "cityInfo" : "?"}

@Produces

{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } }

DELETE

api/address/street

Deleting the address with the given street

api/address/{street}

JSON

@PathParam

{"street"}

@Produces

{ "street": "Street", "additionalInfo": "2nd", "cityInfo": { "zipCode": "?", "city": "?" } }

HobbyAPI

POST

api/hobby	Creating a new hobby
------------------	----------------------

api/hobby/

JSON

@Consumes

{"name" : "?", "description" : "?"}

@Produces

{"name" : "?", "description" : "?"}

GET

api/hobby/all	Return all hobbies with all infos
api/hobby/hobby	Return all addresses with all infos

api/hobby/all

JSON

@Produces

[{"name" : "?", "description" : "?"}, {"name" : "?", "description" : "?"}, {"name" : "?", "description" : "?"},...]

api/hobby/{hobby}

JSON

@PathParam

("hobby")

@Produces

{"name" : "?", "description" : "?"}

PUT

api/hobby	Updating a hobby
------------------	------------------

api/hobby/{hobby}

tilf'j id

JSON

@PathParam

{"hobby"}

@Consumes

{"name" : "?", "description" : "?"}

@Produces

{"name" : "?", "description" : "?"}

DELETE

api/hobby/hobby

Deleting the hobby with the given hobby

api/hobby/{hobby}

JSON

@PathParam

{"hobby"}

@Produces

{"name" : "?", "description" : "?"}

PhoneAPI

POST

api/phone	Creating a new phone
------------------	----------------------

api/phone/

JSON

@Consumes

```
{"number" : "?", "description" : "?", "ie" : "?"}
```

@Produces

```
{"number" : "?", "description" : "?", "ie" : "?"}
```

GET

api/phone/number	Return multiple phones with given phone number with all infos
-------------------------	---

api/phone/all	Return all phones with all infos
----------------------	----------------------------------

api/phone/{number}

JSON

@PathParam

("number")

@Produces

```
{"number" : "?", "description" : "?", "ie" : "?"}
```

api/phone/all

JSON

@Produces

```
[{"number" : "?", "description" : "?", "ie" : "?"}, {"number" : "?", "description" : "?", "ie" : "?"}, {"number" : "?", "description" : "?", "ie" : "?"}, ...]
```

PUT

api/phone

Updating a phone

api/phone/{phone}

tilf'j id

JSON

@PathParam

{"phone"}

@Consumes

{"number" : "?", "description" : "?", "ie" : "?"}

@Produces

{"number" : "?", "description" : "?", "ie" : "?"}

DELETE

api/phone/number

Deleting the phone with the given number

api/phone/{number}

JSON

@PathParam

{"number"}

@Produces

{"number" : "?", "description" : "?", "ie" : "?"}

Test

JUnit test

Som udgangspunkt har vi valgt at teste alle metoder i de forskellige facadeklasser via JUnit. Der er taget udgangspunkt i best-case scenario og derfor tester vi kun metoderne udfra hvordan de skal bruges. Vi har valgt at gøre dette da vi via REST Api'erne kan håndtere brugeres brug af de bagved liggende metoder. Dog blev der implementeret nye metoder i facaderne i slutningen af forløbet som vi ikke har nået at teste.

REST assured


Vi har valgt at teste POST, GET og DELETE på person og company API'erne for at sikre at disse fungerer. Der tages også her kun udgangspunkt i best-case scenario da JSON input kontrolleres via frontend og vi derved har forholdsvis godt styr på det input der kommer.

Resultater


The screenshot displays two JUnit test result windows. The top window, titled 'JUnit.AddressJUnitTest', shows a green progress bar at 100.00% and the text 'All 5 tests passed. (2.358 s)'. The bottom window, titled 'JUnit.CityInfoJUnitTest', also shows a green progress bar at 100.00% and the text 'All 5 tests passed. (2.04 s)'. Both windows include a vertical toolbar on the left with icons for running, pausing, and other test actions.

Test Class	Status	Duration
JUnit.AddressJUnitTest	100.00 %	2.358 s
JUnit.CityInfoJUnitTest	100.00 %	2.04 s






Junit.CompanyJUnitTest ☒



100.00 %




All 7 tests passed. (2.215 s)




>>






Junit.HobbyJUnitTest ☒



100.00 %




All 4 tests passed. (2.036 s)




>>






Junit.PersonJUnitTest ☒



100.00 %




All 6 tests passed. (2.555 s)




>>






Junit.PhoneJUnitTest ☒



100.00 %



All 5 tests passed. (2.026 s)



>>

Hvem og hvad

Dennis – 5 Point

- Person.class, PersonFacade, PersonRestService
- Hobby.class, HobbyFacade, HobbyRestService
- Phone.class, PhoneFacade, PhoneRestService
- Frontend

Andreas – 5 Point

- Company.class, CompanyFacade, CompanyRestService
- CityInfo.class, CityInfoFacade, CityInfoRestService
- Address.class, AddressFacade, AddressRestService
- InfoEntity.class, InfoEntityFacade, InfoEntityRestService
- Test og Exceptions

Implementering af arv

Vi benytter arv ved Person og Company som arver fra InfoEntity. Derved bliver både persons og companies persisteret i samme tabel i databasen, hvilket gør det nemmere at holde dataen samlet og søge efter den. Dog gør dette også at databasen ikke opfylder normalformerne - men disse er blevet tilsidesat i dette projekt da vi benytter JPA.

HowToTest

JUNIT

For at teste via JUNIT gå til Test Package - JUNIT:

{entitet}JUnitTest

- kørs test for at se resultat.

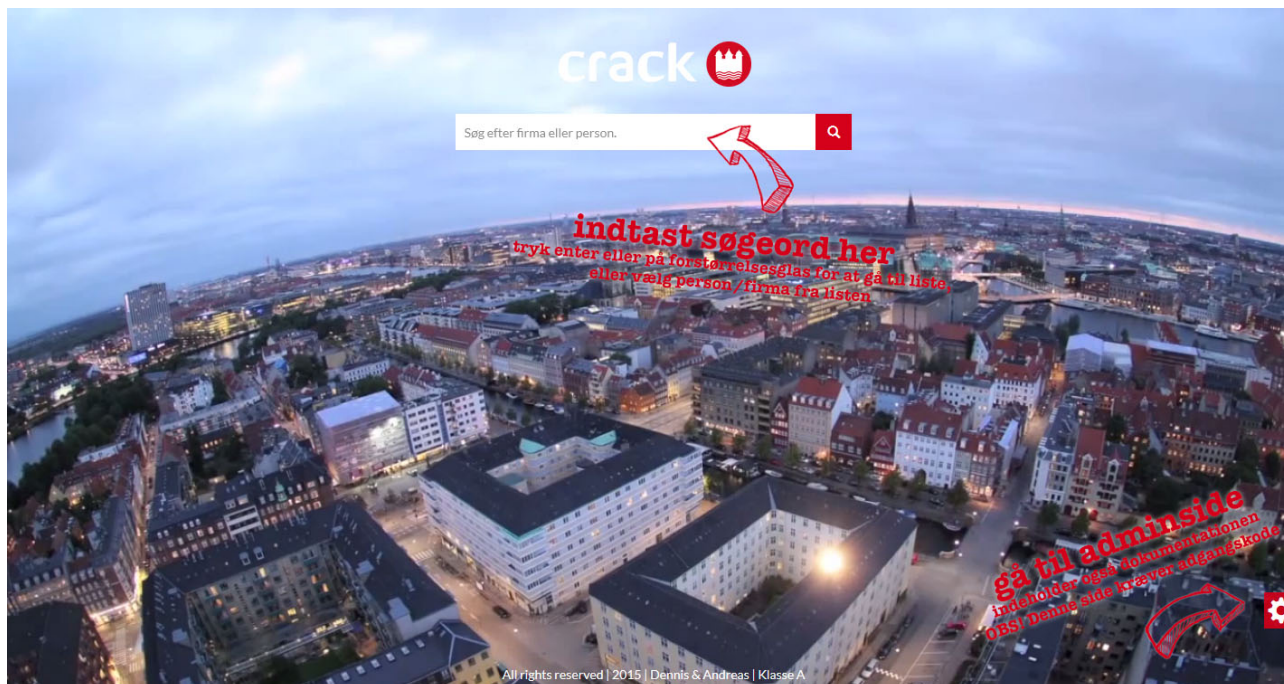
REST assured

For at teste via REST assured gå til Test Package - restapi - CompanyApiTest/PersonApiTest. Kør testen for at se resultat.

POSTMAN

Ved at følge APIet hvori der er eksempel på JSON brug kan de forskellige API'er let testes via PostMan.

Forside



Liste med personer og/eller firmaer

crack

Søg efter firma eller person.

klik på entitet
dette sender dig videre til infoside om
den specifikke entitet

		Email:
	Accumsan Neque Et Incorporated	undefined
	Ad LLP	undefined
	Auctor Incorporated	undefined
	Augue Id Ante Foundation	undefined

indtast søgeord her
tryk enter eller på forstørrelsesglas for at gå til liste,
eller vælg person/firma fra listen

Admin side

Person	Company	Hobby	Phone	Address	City		+	Log out
--------	---------	-------	-------	---------	------	--	---	---------

find dokumentation her

opret ny entitet

Use the list to perform tasks on chosen entities

This page lets you manage all the entities, from here you can create, update and delete entities and persist them in the database

Id				
101	In Consequat Enim Corporation	Update	Delete	
102	Est Mollis Non LLC	Update	Delete	
103	Velit Inc.	Update	Delete	
104	Accumsan Neque Et Incorporated	Update	Delete	
105	Hendrent Id Ltd	Update	Delete	
106	In Aliquet Ltd	Update	Delete	
107	Praesent Consulting	Update	Delete	
108	Phasellus Corporation	Update	Delete	
109	Tellus Incorporated	Update	Delete	
110	In Omare Consulting	Update	Delete	
111	Parturient Montes Corporation	Update	Delete	
112	Luctus Aliquet Odio Institute	Update	Delete	
113	Sed Corp.	Update	Delete	
114	Inceptos Hymenaeos LLP	Update	Delete	
115	Suspendisse Tristique PC	Update	Delete	
116	Enim Associates	Update	Delete	
117	Convallis Convallis Company	Update	Delete	
118	Ipsum Inc.	Update	Delete	
119	Mauris Ut Consulting	Update	Delete	
120	Mollis Foundation	Update	Delete	
121	Rutrum Lorem Corp.	Update	Delete	
122	Eu Erat Sempur LLC	Update	Delete	
123	Faucibus Ut Industries	Update	Delete	

klik for at update entitet,
udfyld data i popup vindue og klik save

klik for at delete entitet