

Test Provider Manual SOAP



It is advised to read the [eHealthBox](#), [consent](#), [therapeutic link](#) or [therapeutic exclusion](#) user manual first to gain a better understanding of the framework.

- [Introduction](#)
- [Target audience](#)
- [Project tree](#)
- [Maintenance](#)
 - [Web](#)
- [Features](#)
 - [Create a feature](#)
 - [Edit a feature](#)
- [Steps](#)
 - [Create a step file](#)
 - [Create a step definition](#)
- [Utilities](#)
 - [Create a utility](#)
 - [Edit a utility](#)
- [FAQ](#)
 - [How to add a new service](#)

Introduction

This manual is meant for everyone that works on the development of the SOAP testing framework.

Target audience

- Developers of the eHealth SOAP services

Project tree

```
src
  main
    java
    resources
      config
        p12
      acc
  test
    java
      be
        imec
          hie
            features
              consent
              ehealthbox
              therapeuticExclusion
              therapeuticLink
            runners
            steps
              consent
              ehealthbox
              therapeuticExclusion
              therapeuticLink
            utils
              consent
              ehealthbox
              therapeuticExclusion
              therapeuticLink
          resources
            attachments
            testdata
            chromedriver.exe
```

Maintenance

Web

The **chromedriver**, found at **src/test/resources/chromedriver.exe**, has to be updated occasionally to keep it up-to-date with the most recent version of Chrome.

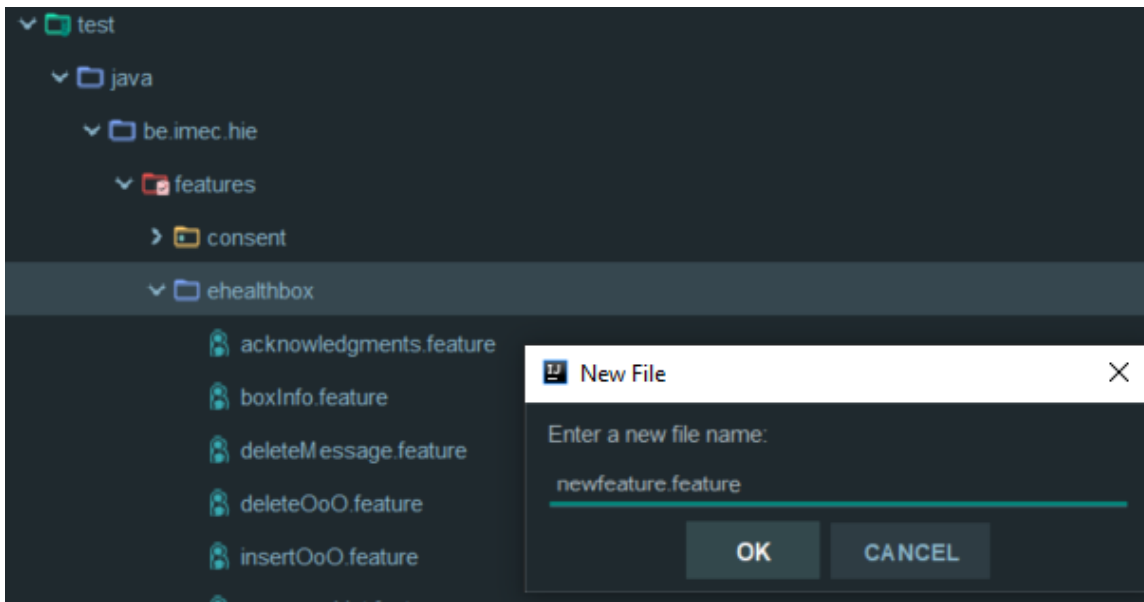
The most recent version of chromedriver can be found [here](#).

Replace to **chromedriver** in the project with the new one from the link above.

Features

Create a feature

Create a new feature file, with a name fitting for the new feature, in the package of an existing service (e.g. features/ehealthbox). For example:



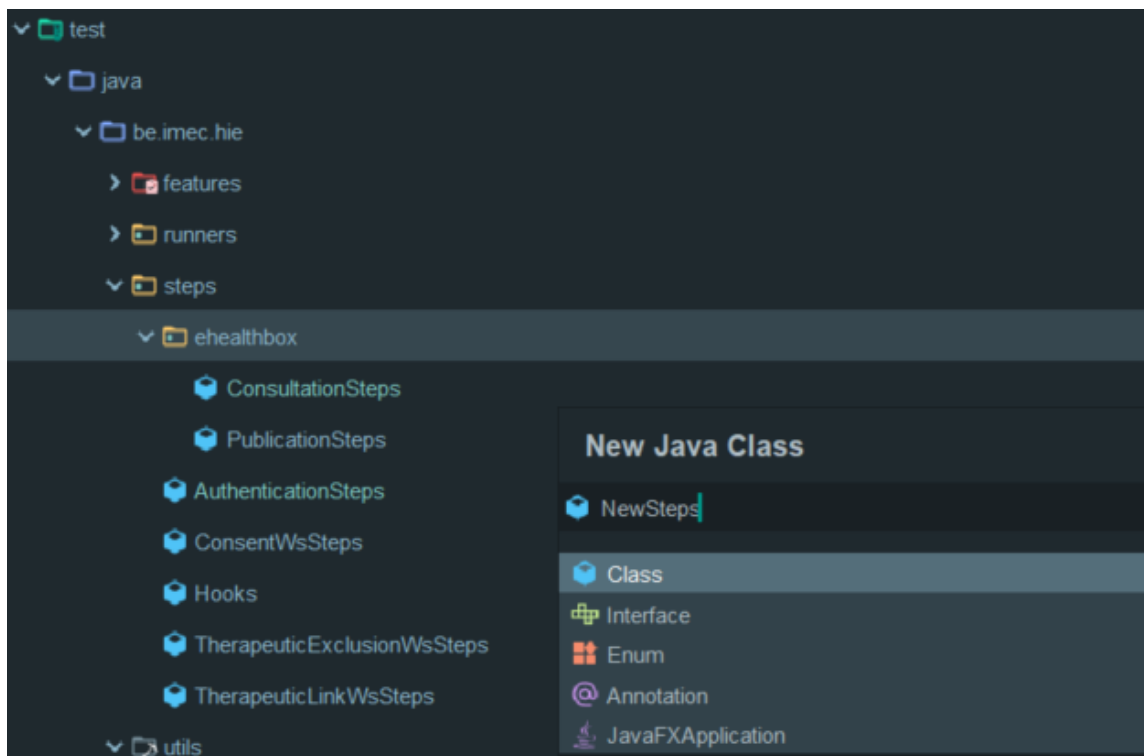
Edit a feature

A very good explanation of everything you can do with a feature file can be found [here](#).

Steps

Create a step file

You can write a new step in one of the existing step files or you can create a new java class for it in the steps package. For example:



Create a step definition

A very good explanation of how step definitions work can be found [here](#).

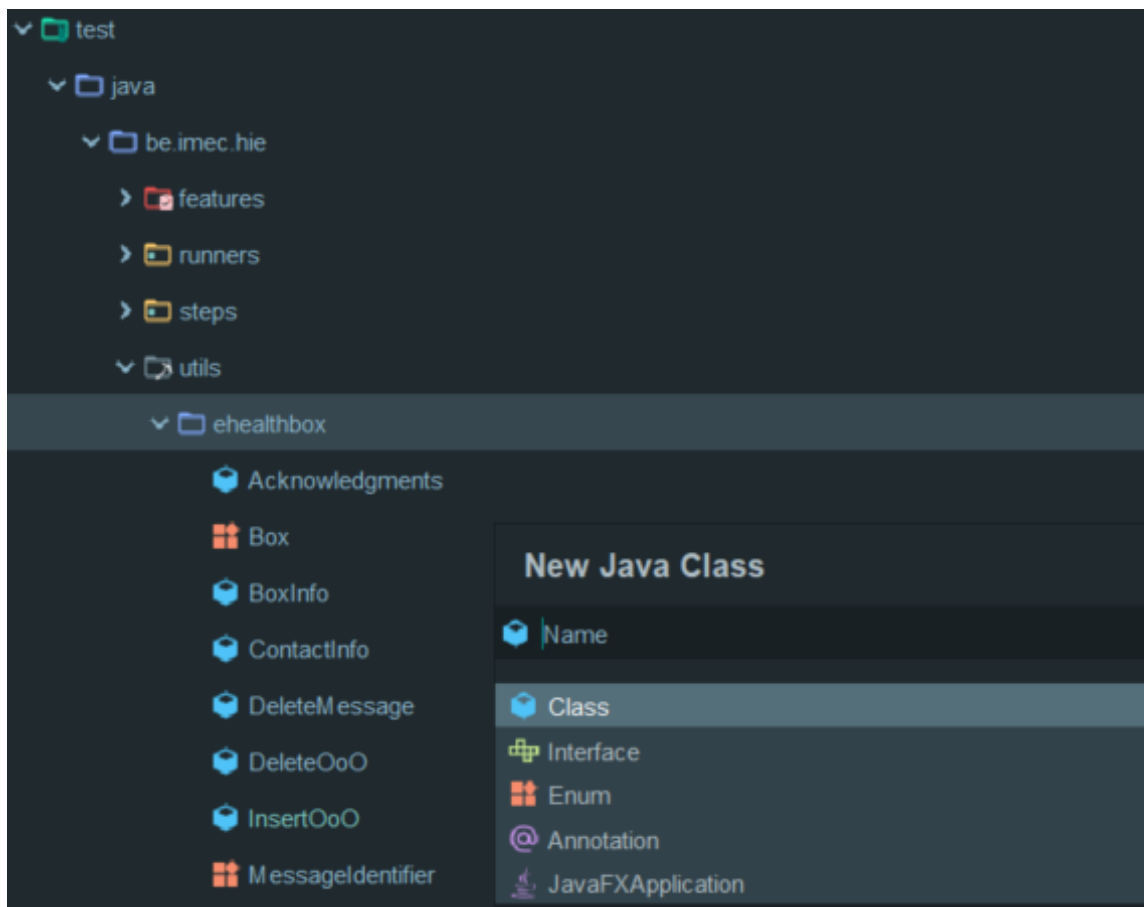
Some useful tips to work with step definitions can be found [here](#).

Utilities

The utilities are java classes that are used to make writing the step definitions easier, used for test validation, etc...

Create a utility

Create a new java class, with a name fitting for the new utility, in the package of an existing service (e.g. utilities/ehealthbox). For example:



Edit a utility

You can write utilities for anything you want in these files.

An example of how the validation of the tests works can be found in, for example, `utils/ehealthbox/BoxInfo.java`.

FAQ

How to add a new service

First of all, you have to create new packages for your new service:

- Create a new package in the feature package (e.g. `features/newservice`)
- Create a new package in the steps package (e.g. `steps/newservice`)
- Create a new package in the utils package (e.g. `utils/newservice`)

When that is done, you can follow these steps in the following order:

1. Create a new feature file
2. Edit your new feature file
3. Create a new step file
4. Create empty step definitions
5. Create a new utility
6. Write validation code for your new utility