# TestSuite description GetCareContact

This document describes the testcases used to verify the implementation of GetCareContact before integration with the national platform.

The testsuite is packaged as a zip-file and contains a SoapUI project, together with some supporting scripts and documents.

## Tools

### SoapUI

The testsuite uses SoapUI to verify the implementation. Documentation of SoapUI can be found at <http://www.soapui.org/>

Link to download site: <http://sourceforge.net/projects/soapui/files/soapui/4.5.2/>

Install SoapUI according to the documentation.

## Setup instructions

1. Unzip the testsuite package.
2. Copy the jar-file ‘soapui-support.jar’ to <SoapUI install dir>/bin/ext
3. Open SoapUI and import the testsuite project, choose ‘Import Project’ from the File-menu.
4. If your WebService endpoint requires a SSL Certificate, this can be configured from the Preferences (in the File menu). In the Preferences window open the ‘SSL Settings’ tab and import the Keystore containing the Client Certificate.
5. Update test-data to match the contents in your system.
6. You should now be able to run the test suite!

## Testcase description

### Date boundaries

This testcase verifies filtering with date intervals.

The testcase will require a patient with at least nine CareContacts:

1. A contact with DocumentTime within the given range, and the CareContactTimePeriod (both start and end) before or after the given range.
2. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts before and ends within the given range.
3. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts within and ends after the given range.
4. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts before and ends after the given range.
5. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts within and ends within the given range.
6. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts before and no end time.
7. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts before and ends before the given range.
8. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts after and ends after the given range.
9. A contact with DocumentTime outside the given range and a CareContactTimePeriod that starts after and no end time.

The CareContactId's of case 1-6 should be added as a comma-separated list in the expectedCareContactIds field.

The CareContactId's of case 7, 8 and 9 should be added as a comma-separated list in the unexpectedCareContactIds field.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### CareUnitHSAid Filter

This testcase verifies filtering regarding to CareUnit, the response should only contain information with the correct CareUnitHSAid for the given patient.

The patient should have several CareContacts, with at least one CareContact from the given CareUnitHSAid.

The CareContactId's of the expected CareContacts should be added as a comma-separated list in the expectedCareContactIds field.

The CareContactId's of CareContacts from other CareUnit's should be added as a comma-separated list in the unexpectedCareContactIds field.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Only PatientId

Verifies that the result only contains information with correct PatientId.

Enter a PatientId that is stored in the source system.

The CareContactId's of the expected CareContacts should be added as a comma-separated list in the expectedCareContactIds field.

The CareContactId's of CareContacts from other CareUnit's should be added as a comma-separated list in the unexpectedCareContactIds field.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Non-existing PatienId

Verifies that the service returns an empty result instead of a Soap Fault if a non-existing PatientId is given.

Enter a PatientId that is not found in the source system.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Only CareContactId

Verifies that the result only contains information for a single CareContactId.

Enter a CareContactId that is stored in the source system.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Non-existing CareContactId

Verifies that the service returns an empty result instead of a Soap Fault if a non-existing CareContactId is given.

Enter a careContactId and sourceSystemHSAid that is not found in the source system.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Http-header Filter

Verifies that the result only contains information with correct PatientId, and that the source system should filter CareContacts based on the HttpHeader 'x-rivta-original-serviceconsumer-hsaid'.

Enter a PatientId that has more than one CareContact in the source system.

Enter a HSAid in the field httpHeaderHsaId, that should have the samve value as some of the patients CareContacts.

Add one or more CareContactId's of contacts that should be included in the response based on the filtering rules to the field 'expectedCareContactIds'.

Add one or more CareContactId's of contacts that should not be included in the response based on the filtering rules to the field 'unexpectedCareContactIds'.

The field logicalAddress should contain a value that does not affect the list of CareContacts.

### Fault testcase SourceSystemHSAid and CareContactId

The test verifies that the source system returns a Soap Fault if the sourceSystemHSAid

is empty when careContactId has a value.

Enter a CareContactId of a CareContact that should be found in the source system.

The fields httpHeaderHsaId and logicalAddress should contain values that does not affect the list of CareContacts.

### Fault testcase SourceSystemHSAid and LogicalAddress

The test verifies that the source system returns a Soap Fault if the logicalAddress and sourceSystemHSAid doesn't match.

Enter a CareContactId and the correct SourceSystemHSAid of a CareContact that should be found in the source system.

The field logicalAddress should not match SourceSystemHSAid.

The field httpHeaderHsaId should contain a value that does not affect the list of CareContacts.