

# Claim Portal: A2A Token Implementation Workshop

Stephen Goodwin  
Head of Product, MIB (CPL)

Tuesday 18<sup>th</sup> October 2022

**CLAIMS**PORTAL LTD





## Agenda / Housekeeping

- Timings for today's session
- Introductions to the teams today
- We will be recording the session
- How to ask questions during the session – Slido (see link in Teams chat)
- Q&A's at end of session / follow up questions after the event
- Claims Portal Support
- Background / What we will cover



# Introductions

## **Claims Portal Ltd:**

- Sue Brown - Director
- Martin Ward - Director

## **CPL Service Team:**

- Stephen Goodwin – Head of Product
- Alison Gibbs – Product Manager
- Anita Mehta – Product Support Lead
- Warren Sibley – Customer Services Assistant

## **CRIF (CPL Technology Partner):**

- Maurizio Castelli – ITSM
- Diego Frezzato – Project Coordinator
- Ileana Magno – Senior Business Analyst
- Nevia McKiernan – Senior Account Manager
- Michele Modestino – Junior Business Analyst
- Vincenzo Vitti – Development Lead



# Claims Portal Support

## Claims Portal Operational Team:

- CPL Helpdesk – First line support
- CPL Service Team – Second line support, Audit, Projects Team
- Claims Portal Directors – Referral and escalation point

## Customer Points of Contact:

- Claims Portal Website - [Claims Portal | Home](#)
- Claims Portal Helpdesk – [helpdesk@rapidclaimsettlement.org.uk](mailto:helpdesk@rapidclaimsettlement.org.uk)
- Claims Portal Email - [ClaimsPortal@mib.org.uk](mailto:ClaimsPortal@mib.org.uk)



## Claims Portal Technology Partner - CRIF

- Development and maintenance of all Claims Portal Applications – Production, Integration and UAT (User Acceptance Testing)
- A2A Requests
- Insurer Index uploads
- Technical Specifications (API Integration)
- Projects
- Releases
- Based in Bologna, Italy





## Background / What we will cover

Claims Portal Ltd will be introducing a token as part of the A2A messaging. The token will apply an additional layer of security to verify the identity of the user, along with current credentials.

Draft Technical Specifications have been released and are available on our website for A2A users to plan and assess the changes required.

Today, we would like to cover the following:

- An overview of the A2A token messaging
- Technical details on the system changes required
- Use of the integration site for testing
- Details of the support available to implement the changes ahead of the agreed go live date
- Client ID's
- Q&A session

# Claims Portal

## A2A Token and Password Change





## Agenda

- Schedule
- A2A Token introduction
- getToken()
- refreshToken()
- changePassword()
- Examples of A2A requests/responses
- Error handling
- Preparation to go live










# Schedule

Months		Oct					Nov				Dec				Jan						
Row ID	WBE/WBE	03/10/22	10/10/22	17/10/22	24/10/22	31/10/22	07/11/22	14/11/22	21/11/22	28/11/22	05/12/22	12/12/22	19/12/22	26/12/22	02/01/23	09/01/23	16/01/23	23/01/23	30/01/23		
77.3	A2A Acceptance Testing	◆	◆	→				◆													
77.5	A2A Integration					◆	→										◆				
77.6	Production																	◆			

**Legenda**

-  Elapsed total
-  Release of the **Test Case Book**
-  Release in **UAT/INT**
-  Release in **PROD**
-  Holiday/Freezing period



## A2A Token introduction

To enhance the security of the A2A service the login mechanism on the Claims Portal A2A has been evolved from a standard “Username+AsUser+Password” to a Token-based Authentication.

This involves the following changes to the system:

- Introduction of the **getToken()** functionality
- Introduction of the **refreshToken()** functionality
- Introduction of the **changePassword()** functionality
- Changes to the existing methods with the replacement of the “Username+AsUser+Password” login with the token-based authentication.



## New naming convention

Please note that the following existing input field have been renamed as follows:

OLD naming convention	NEW naming convention
username	userID
asUser	userAsID

In addition, a new namespace has been introduced:

- FROM xmlns:ws=<http://ws.pip.crif.com/> TO xmlns:wstk=<http://wstk.pip.crif.com/>
- FROM “ws” TO “wstk”



# getToken()

**clientID** = attribute used to uniquely identify the software house that is performing the getToken()/refreshToken() on behalf of the CR/COMP

getToken():

- userID
- userAsID
- Password
- clientID

accessToken validity = 2h (7200 sec)  
refreshToken validity = 24h (86400 sec)

getTokenResponse():

- accessToken
- refreshToken
- accessTokenExpiresIn
- refreshTokenExpiresIn

addClaim():

- accessToken
- addClaim xml

The accessToken must be included in all the requests

addClaimResponse():  
Claim successfully added on the system



## getToken()

If invoked during the validity period of the accessToken (2 hours) the getToken() will return the same accessToken and the same refreshToken, e.g.:

Time	getToken() details	getToken() response
T <sub>0</sub>	userID=user1 userAsID=userAsID1 password=password clientId=clientID1	accessToken refreshToken accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 sec
T <sub>1</sub> (after 1h 45 min)	userID=user1 userAsID=userAsID1 password=password clientId=clientID1	accessToken refreshToken accessTokenExpiresIn=900 sec refreshTokenExpiresIn=80100 sec

In order to refresh the accessToken the refreshToken() **must** be used. The refreshToken() can be used both before and after the expiration of the accessToken.



## getToken()

The accessToken is strictly linked to the userID and userAsID. If you use more than one userAsID you need to perform the getToken() call for each userAsID.

Time	getToken() details	getToken() response
T <sub>0</sub>	userID=user1 userAsID=userAsID1 password=password clientId=clientID1	accessToken1 refreshToken1 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 sec
T <sub>1</sub> (after 15 min)	userID=user1 userAsID=userAsID2 password=password clientId=clientID1	accessToken2 refreshToken2 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 sec

Please refer to the technical specifications for the correct handling of tokens for different userAsIDs for the **GetNotificationsList()** and **RemoveNotification()** methods.



# refreshToken()

getClaim():

- accessToken
- getClaim xml

clientID = attribute used to uniquely identify the software house that is performing the getToken()/refreshToken() on behalf of the CR/COMP

getClaimresponse(): claim details successfully retrieved

refreshToken():

- userID
- userAsID
- clientID
- refreshToken

accessToken validity = 2h  
refreshToken validity = 24h – the time passed from the previous getToken()

refreshTokensponse():

- new accessToken
- new refreshToken
- new accessTokenExpiresIn
- new refreshTokenExpiresIn

addInterimSPFResponse():

- new accessToken
- addInterimSPFResponse xml

The new accessToken **must** be included in all following requests

addInterimSPFResponseResponse():  
addInterimSPFResponse successfully sent to CR



# refreshToken()

Time	Request details	Response details
$T_0$	getToken(): userID=user1 userAsID=userAsID1 password=password clientId=clientId1	getTokenResponse(): accessToken=accessToken1 refreshToken=refreshToken1 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 sec
$T_1$ (after 1h 45min)	refreshToken(): userID=user1 userAsID=userAsID1 clientId=clientId1 refreshToken=refreshToken1	refreshTokenResponse(): accessToken=accessToken2 refreshToken=refreshToken2 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=80100 sec
$T_2$	refreshToken(): userID=user1 userAsID=userAsID1 clientId=clientId1 refreshToken=refreshToken1	BAD TOKEN ERROR
$T_3$ (after 2h $T_0$ )	refreshToken(): userID=user1 userAsID=userAsID1 clientId=clientId1 refreshToken=refreshToken2	refreshTokenResponse(): accessToken=accessToken3 refreshToken=refreshToken3 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=79200 sec

Only the last received refreshToken can be used when invoking the refreshToken()





# refreshToken()

Time	Request details	Response details
T <sub>0</sub>	getToken(): userID=user1 userAsID=userAsID1 password=password clientId=clientID1	getTokenResponse(): accessToken=accessToken1 refreshToken=refreshToken1 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 sec
T <sub>1</sub>	getToken(): userID=user1 userAsID=userAsID2 password=password clientId=clientID1	getTokenResponse(): accessToken=accessToken2 refreshToken=refreshToken2 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=86400 ses
T <sub>2</sub> (after 1h 30min T <sub>0</sub> )	refreshToken(): userID=user1 userAsID=userAsID2 clientId=clientID1 refreshToken=refreshToken2	refreshTokenResponse(): accessToken=accessToken3 refreshToken=refreshToken3 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=81000 sec
T <sub>3</sub> (after 1h 45min T <sub>0</sub> )	refreshToken(): userID=user1 userAsID=userAsID1 clientId=clientID1 refreshToken=refreshToken2	BAD TOKEN ERROR
T <sub>4</sub> (after 1h 45min T <sub>0</sub> )	refreshToken(): userID=user1 userAsID=userAsID1 clientId=clientID1 refreshToken=refreshToken1	refreshTokenResponse(): accessToken=accessToken4 refreshToken=refreshToken4 accessTokenExpiresIn=7200 sec refreshTokenExpiresIn=80100 sec

As per the accessToken the refreshToken is strictly linked to the userID and userAsID used to obtain it

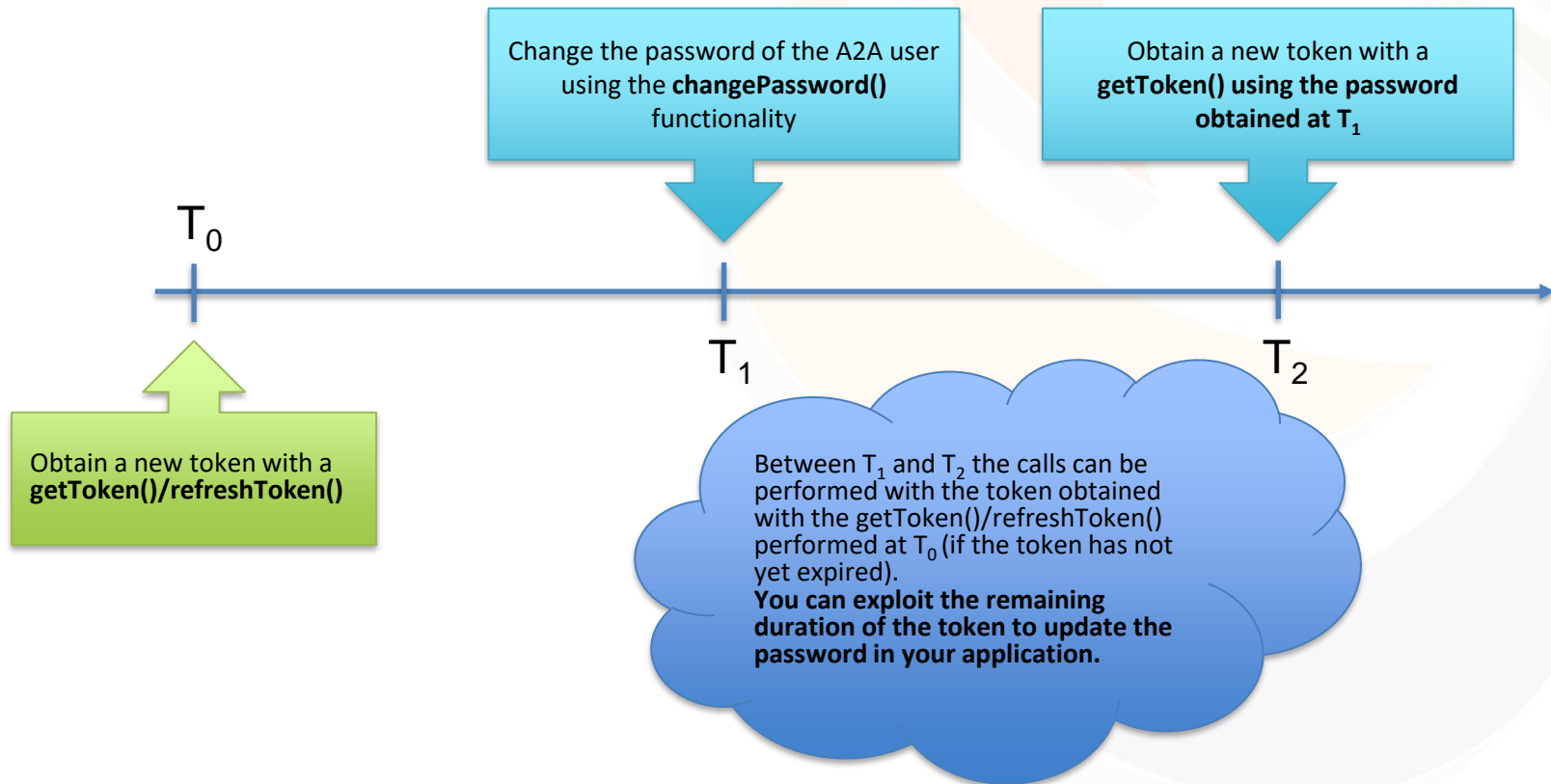


## changePassword()

- A2A password validity: **90 days**
- New A2A functionality to change the A2A user's password: **changePassword()**
- **It is possible to change the password even if the token is no longer valid** this reduces the risk of login errors.
- After the change password the next getToken() must be performed using the new password.
- It is not possible to change the password of the web users, the **changePassword()** method is valid for the A2A users only.
- New validation rules for the password, please refer to the technical specifications for further details



## changePassword()





## changePassword()

Please note:

- Using the same functionality it is also possible to change the password when the current password has already expired. Our recommendation is to concentrate your tests on changing the A2A password when the password is still valid, this replicates the ideal live scenario when the password needs to be changed before the expiration date to avoid any login issue.
- If you need to test the change password when the password has already expired, please complete the “a2a Contact Use” form, attach to an email and send to the Helpdesk with the following in the subject header: “*a2a token test [Integration] Change expired password*” ensure that you complete the information requested in the form, as we need your: Claims Portal organisationID, organisationName and the A2A user’s username.
- Upon first access with new A2A credentials (or with a new password created by the Helpdesk) it is mandatory to set a new password, trying to perform a getToken() or a refreshToken() before changing the password will result in the message “Please set a new password” appearing.



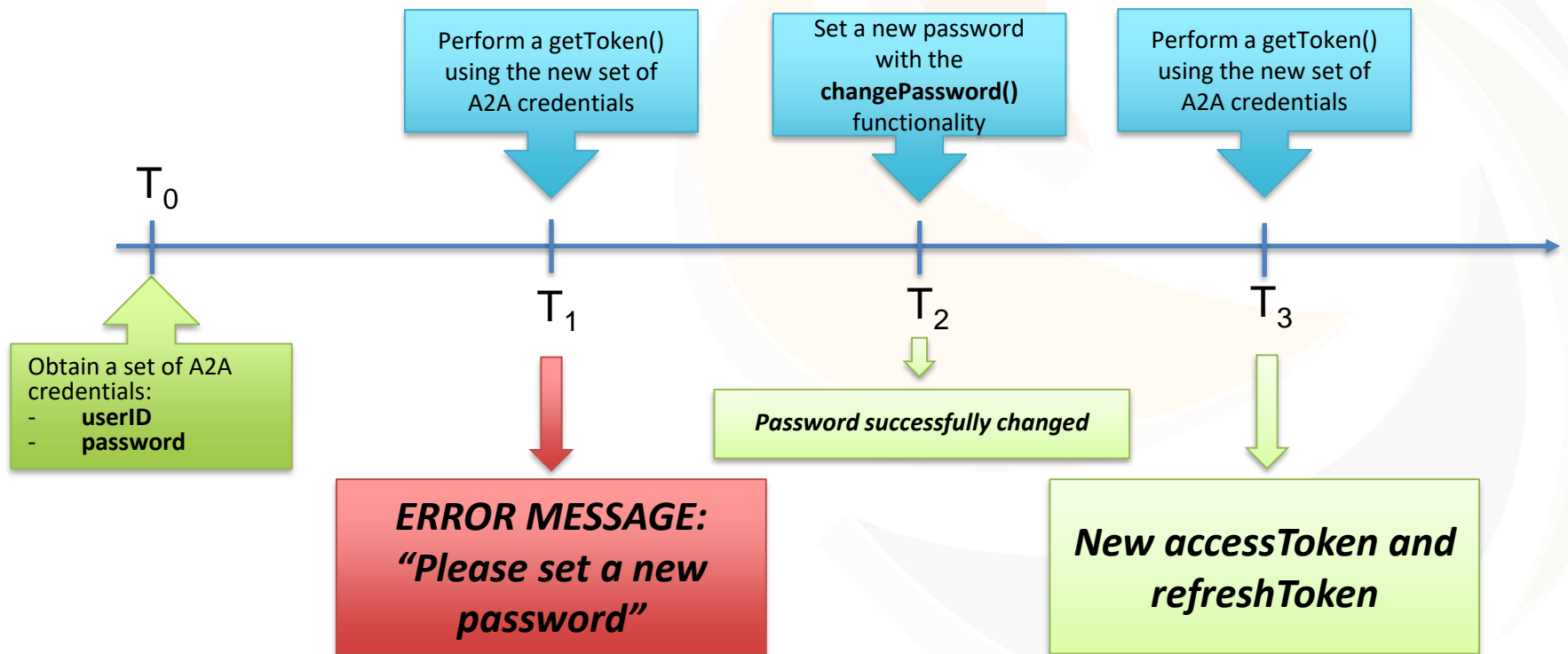
## changePassword()

**Please be aware that as A2A users you are responsible for ensuring that the A2A password is changed within the 90 days specified in the technical documentation.**

**If you are a software house you are responsible for ensuring that your clients password is changed within the 90 days specified in the technical documentation.**



# First A2A login - changePassword()





## Request/responses examples – getToken()

Following, are some examples of the new requests/responses involving the new Token:

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:getToken&gt;       &lt;userID&gt;a2auserID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;       &lt;password&gt;a2aPassword&lt;/password&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;     &lt;/wstk:getToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:getTokenResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;tokenResponse&gt;         &lt;accessToken&gt;0JyjKdKfM/YyWVT2MieBAtmTqomwYnzBFMUK5Zfm3uh0QE TJEU6i+A==&lt;/accessToken&gt;         &lt;refreshToken&gt;K+3P2j7TiT0kI7Bi7mAfLJqbDAuwz058H6BpuI5IMEX1S aa5pXpxhQ==&lt;/refreshToken&gt;         &lt;accessTokenExpiresIn&gt;7200&lt;/accessTokenExpiresIn&gt;         &lt;refreshTokenExpiresIn&gt;86400&lt;/refreshTokenExpiresIn&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:getTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



# searchClaim()

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:searchClaims&gt;       &lt;searchClaimCriteria&gt;         &lt;phaseCacheId&gt;ClaimSubmitted&lt;/phaseCacheId&gt;         &lt;sortField&gt;ApplicationId&lt;/sortField&gt;         &lt;sortOrder&gt;Desc&lt;/sortOrder&gt;       &lt;/searchClaimCriteria&gt;       &lt;accessToken&gt;0JyjKdkFM/YyWVT2MieBAtmTqomwYnzBFMUK5Zfm3uh0QETJEU6i+A==&lt;/accessToken&gt;     &lt;/wstk:searchClaims&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:searchClaimsResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;claimInfosListResponse&gt;         &lt;code&gt;Ok&lt;/code&gt;         &lt;claimsList&gt;           &lt;activityEngineGuid&gt;96AC761A-2FFF-11BC-9789-005056826327&lt;/activityEngineGuid&gt;           &lt;applicationId&gt;000000000016176&lt;/applicationId&gt;           &lt;applicationReferences&gt;CDPDSurname, CDPDName CDPDMiddle&lt;/applicationReferences&gt;           &lt;attachmentsCount&gt;0&lt;/attachmentsCount&gt;           &lt;creationTime&gt;2022-09-06T16:02:45.377+02:00&lt;/creationTime&gt;           &lt;currentUserID&gt;CurrentUserID&lt;/currentUserID&gt;           &lt;lockStatus&gt;1&lt;/lockStatus&gt;           &lt;phaseCacheId&gt;ClaimSubmitted&lt;/phaseCacheId&gt;           &lt;phaseCacheName&gt;Claim Submitted&lt;/phaseCacheName&gt;           &lt;printableDocumentsCount&gt;2&lt;/printableDocumentsCount&gt;           &lt;versionMajor&gt;63&lt;/versionMajor&gt;           &lt;versionMinor&gt;3&lt;/versionMinor&gt;         &lt;/claimsList&gt;       &lt;/claimInfosListResponse&gt;     &lt;/ns2:searchClaimsResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>





# refreshToken()



~2 hours later

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:refreshToken&gt;       &lt;userID&gt;a2auserID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;       &lt;refreshToken&gt;K+3P2j7TiT0kI7Bi7mAfLJqbDAuwz058H6       BpuI5IMEX1Saa5pXpxhQ==&lt;/refreshToken&gt;     &lt;/wstk:refreshToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:refreshTokenResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;tokenResponse&gt;         &lt;accessToken&gt;3GB9RoEBEnPU9zVANAIgW11ncVePyD01MfvYzexPj0eLHyU         kWDL7hg==&lt;/accessToken&gt;         &lt;refreshToken&gt;YKtHykk8+yZICuX8p9rFvJp0+vyH9pOH/ngTKb6sjzof/y         9fFLG9SQ==&lt;/refreshToken&gt;         &lt;accessTokenExpiresIn&gt;7200&lt;/accessTokenExpiresIn&gt;         &lt;refreshTokenExpiresIn&gt;80100&lt;/refreshTokenExpiresIn&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:refreshTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



# searchClaim()

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:searchClaims&gt;       &lt;searchClaimCriteria&gt;         &lt;phaseCacheId&gt;ClaimSubmitted&lt;/phaseCacheId&gt;         &lt;sortField&gt;ApplicationId&lt;/sortField&gt;         &lt;sortOrder&gt;Desc&lt;/sortOrder&gt;       &lt;/searchClaimCriteria&gt;       &lt;accessToken&gt;3GB9RoEBEnPU9zVANAiGw11ncVePyD01MfvYzexPjOeLHyUkwDL7hg==&lt;/accessToken&gt;     &lt;/wstk:searchClaims&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;           </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:searchClaimsResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;claimInfosListResponse&gt;         &lt;code&gt;0K&lt;/code&gt;         &lt;claimsList&gt;           &lt;activityEngineGuid&gt;96AC761A-2FFF-11BC-9789-005056826327&lt;/activityEngineGuid&gt;           &lt;applicationId&gt;000000000016176&lt;/applicationId&gt;           &lt;applicationReferences&gt;CDPDSurname, CDPDName CDPDMiddle&lt;/applicationReferences&gt;           &lt;attachmentsCount&gt;0&lt;/attachmentsCount&gt;           &lt;creationTime&gt;2022-09-06T16:02:45.377+02:00&lt;/creationTime&gt;           &lt;currentUserID&gt;CurrentUserID&lt;/currentUserID&gt;           &lt;lockStatus&gt;1&lt;/lockStatus&gt;           &lt;phaseCacheId&gt;ClaimSubmitted&lt;/phaseCacheId&gt;           &lt;phaseCacheName&gt;Claim Submitted&lt;/phaseCacheName&gt;           &lt;printableDocumentsCount&gt;2&lt;/printableDocumentsCount&gt;           &lt;versionMajor&gt;63&lt;/versionMajor&gt;           &lt;versionMinor&gt;3&lt;/versionMinor&gt;         &lt;/claimsList&gt;       &lt;/claimInfosListResponse&gt;     &lt;/ns2:searchClaimsResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt;           </pre>



# searchClaim() with old token



>2 hours later

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:searchClaims&gt;       &lt;searchClaimCriteria&gt;         &lt;phaseCacheId&gt;ClaimSubmitted&lt;/phaseCacheId&gt;         &lt;sortField&gt;ApplicationId&lt;/sortField&gt;         &lt;sortOrder&gt;Desc&lt;/sortOrder&gt;       &lt;/searchClaimCriteria&gt;       &lt;accessToken&gt;0TyjKDKFM/YyKVT2MieBAtmTqomwYnzBF MUK5Zfm3uh0QETJ Eu6i+A==&lt;/accessToken&gt;     &lt;/wstk:searchClaims&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;           </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:searchClaimsResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;claimInfosListResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Bad Token&lt;/message&gt;         &lt;trace&gt;202209061812380009651542843030&lt;/trace&gt;       &lt;/claimInfosListResponse&gt;     &lt;/ns2:searchClaimsResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt;           </pre>

Error type: 200



## refreshToken() with wrong details (e.g. old refresh token)



>24 hours later

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:refreshToken&gt;       &lt;userID&gt;a2userID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;       &lt;refreshToken&gt;K+3P2j7TiT0kI7Bi7mAfLJqbDAuwz058H6       BpuI5IMEX1Saa5pXpxhQ==&lt;/refreshToken&gt;     &lt;/wstk:refreshToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:refreshTokenResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;tokenResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Bad Token&lt;/message&gt;         &lt;trace&gt;202209061552100002858879041207&lt;/trace&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:refreshTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>

Error type: 200



~24 hours later

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:getToken&gt;       &lt;userID&gt;a2auserID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;     &lt;password&gt;a2aPassword&lt;/password&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;     &lt;/wstk:getToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:getTokenResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;tokenResponse&gt;         &lt;accessToken&gt;e2/nZr9QB2wVbsr19EbnuK68166wk4SrNA1Gs21x48+oo/q CD457Vg==&lt;/accessToken&gt;         &lt;refreshToken&gt;Fes8jthcpyth+zKR5w0IZVwNqkZraZREEjnoyOM2X+RsZt B0k87C1A==&lt;/refreshToken&gt;         &lt;accessTokenExpiresIn&gt;7200&lt;/accessTokenExpiresIn&gt;         &lt;refreshTokenExpiresIn&gt;86400&lt;/refreshTokenExpiresIn&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:getTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



## getToken() with wrong details (e.g. wrong password)

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:getToken&gt;       &lt;userID&gt;a2userID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;       &lt;password&gt;a2aPassword&lt;/password&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;     &lt;/wstk:getToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:getTokenResponse xmlns:ns2="http://wstk.pip.crif.com/"       &lt;tokenResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Login Failed or Expired Credential&lt;/message&gt;         &lt;trace&gt;202209061552090008788338455673&lt;/trace&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:getTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>

Error type: 200



## getToken() with first issued password

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.elpl.crif.com/"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:getToken&gt;       &lt;userID&gt;a2userID&lt;/userID&gt;       &lt;userAsID&gt;userAsID&lt;/userAsID&gt;       &lt;password&gt;firstA2APassword&lt;/password&gt;       &lt;clientID&gt;SH0001&lt;/clientID&gt;     &lt;/wstk:getToken&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:getTokenResponse xmlns:ns2="http://wstk.elpl.crif.com/"&gt;       &lt;tokenResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Please set a new password&lt;/message&gt;         &lt;trace&gt;202210040940510000407329221674&lt;/trace&gt;       &lt;/tokenResponse&gt;     &lt;/ns2:getTokenResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>

Error type: 200



# changePassword()

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:changePassword&gt;       &lt;userID&gt;a2auserID&lt;/userID&gt;       &lt;oldPassword&gt;OldPassword&lt;/oldPassword&gt;       &lt;newPassword&gt;NewPassword&lt;/newPassword&gt;     &lt;/wstk:changePassword&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt; </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:changePasswordResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;stringResponse&gt;         &lt;code&gt;Ok&lt;/code&gt;         &lt;value&gt;Success&lt;/value&gt;       &lt;/stringResponse&gt;     &lt;/ns2:changePasswordResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>





## changePassword() with wrong details (e.g. wrong password)

REQUEST	RESPONSE
<pre> &lt;soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wstk="http://wstk.pip.crif.com/"&gt;   &lt;soapenv:Header/&gt;   &lt;soapenv:Body&gt;     &lt;wstk:changePassword&gt;       &lt;userID&gt;A2AuserID&lt;/userID&gt;       &lt;oldPassword&gt;OldPassword&lt;/oldPassword&gt;       &lt;newPassword&gt;NewPassword&lt;/newPassword&gt;     &lt;/wstk:changePassword&gt;   &lt;/soapenv:Body&gt; &lt;/soapenv:Envelope&gt;           </pre>	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:changePasswordResponse xmlns:ns2="http://wstk.epl.crif.com/"&gt;       &lt;stringResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Invalid or Expired Credential&lt;/message&gt;         &lt;trace&gt;202209061822360004784145253141&lt;/trace&gt;       &lt;/stringResponse&gt;     &lt;/ns2:changePasswordResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt;           </pre>

Error type: 200



## Error Handling

Errors returned by the system can be categorized into two main types.

- **HTTP Error 500 (soap fault)**: in case of unexpected error, the system responds with a soap fault.
- **HTTP Response 200 (ok)**: in case of application failure, the system responds with an error message in the xml response (not a soap fault).

For all error types the system returns a *Trace* that is the error's unique reference on system's logs. If a call is opened with the HelpDesk the Trace information must be communicated so Helpdesk are able to collect the necessary information on the issue.



## Other types of errors (200):

REQUEST	RESPONSE
Get claim with a wrong applicationID	<pre>&lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:getClaimResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;stringResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;ApplicationID Invalid&lt;/message&gt;         &lt;trace&gt;202209061600140000002627379154&lt;/trace&gt;       &lt;/stringResponse&gt;     &lt;/ns2:getClaimResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt;</pre>



## Other types of errors (200):

REQUEST	RESPONSE
Operation performed when not allowed	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:sendLiabilityDecisionResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;claimInfoResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;The operation could not be applied in this point&lt;/message&gt;         &lt;trace&gt;202209061826180004415743096357&lt;/trace&gt;       &lt;/claimInfoResponse&gt;     &lt;/ns2:sendLiabilityDecisionResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



## Other types of errors (200):

REQUEST	RESPONSE
Missing mandatory field	<pre> &lt;env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;ns2:addCPPFRequestResponse xmlns:ns2="http://wstk.pip.crif.com/"&gt;       &lt;claimInfoResponse&gt;         &lt;code&gt;Failure&lt;/code&gt;         &lt;message&gt;Input XML not valid&lt;/message&gt;         &lt;details&gt;org.xml.sax.SAXParseException; cvc- complex-type.4: Attribute 'GrossValueClaimed' must appear on element 'LastClaimantOffer'.&lt;/details&gt;         &lt;trace&gt;202209061600540000897969142122&lt;/trace&gt;       &lt;/claimInfoResponse&gt;     &lt;/ns2:addCPPFRequestResponse&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



## Other types of errors (500):

REQUEST	RESPONSE
Unexpected error	<pre> &lt;env:Envelope xmlns:env="HTTP://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;env:Header/&gt;   &lt;env:Body&gt;     &lt;env:Fault&gt;       &lt;faultcode&gt;soap:Server&lt;/faultcode&gt;       &lt;faultstring&gt;Operation failed, Trace: 202208181234400001569345240180&lt;/faultstring&gt;     &lt;/env:Fault&gt;   &lt;/env:Body&gt; &lt;/env:Envelope&gt; </pre>



## Preparation to Go Live



## Preparation to Go live – A2A Token and Password Change

24<sup>th</sup> January: GO LIVE DATE



NO PROCESS VERSION UPDATE



**No retro-compatibility issues**





## Technical Documentation

- **TECH SPECS - DRAFT A2A interface:** describes the I/O of functions (WSDL file)
- **Tech Specs A2A interface final version release date 31st October**
- **A2A Release Notes DRAFT version :** describes the changes to the A2A
- **A2A Release Note final version release date 31st October**

Available on web site

<https://www.claimsportal.org.uk/developer/application-to-application-a2a-token-authentication-log-in-access/>



## Integration endpoints parallel run

New endpoints and old ones are available in parallel from 07th of November.

	OLD ENDPOINT	NEW ENDPOINT
RTA	<a href="https://piptesta2a.crif.com/PIP.WS/PIPWS">https://piptesta2a.crif.com/PIP.WS/PIPWS</a>	<a href="https://piptesta2a.crif.com/PIP.WSTK/PIPWSTK">https://piptesta2a.crif.com/PIP.WSTK/PIPWSTK</a>
ELPL	<a href="https://piptesta2a.crif.com/ELPL.WS/ELPLWS">https://piptesta2a.crif.com/ELPL.WS/ELPLWS</a>	<a href="https://piptesta2a.crif.com/ELPL.WSTK/ELPLWSTK">https://piptesta2a.crif.com/ELPL.WSTK/ELPLWSTK</a>

The web URL has not changed:

→ <https://piptesta2a.crif.com/>



## Production endpoints parallel run

New endpoints and old ones are available in parallel from 24th of January until 07<sup>th</sup> of February 2023.

	OLD ENDPOINT	NEW ENDPOINT
RTA	<a href="https://www.rapidclaimsettlement.org.uk/PIP.WS/PIPWS">https://www.rapidclaimsettlement.org.uk/PIP.WS/PIPWS</a>	<a href="https://www.rapidclaimsettlement.org.uk/PIP.WSTK/PIPWSTK">https://www.rapidclaimsettlement.org.uk/PIP.WSTK/PIPWSTK</a>
ELPL	<a href="https://www.rapidclaimsettlement.org.uk/ELPL.WS/ELPLWS">https://www.rapidclaimsettlement.org.uk/ELPL.WS/ELPLWS</a>	<a href="https://www.rapidclaimsettlement.org.uk/ELPL.WSTK/ELPLWSTK">https://www.rapidclaimsettlement.org.uk/ELPL.WSTK/ELPLWSTK</a>

The web URL has not changed:

→ <https://www.rapidclaimsettlement.org.uk/>

**NOTE: parallel running in PROD does not mean that you should duplicate the calls. Using the old endpoint allows you to work when you have an issue with your A2A token API.**



## Integration sites and testing

The test site is available for users to test their development and changes to their API. It is not designed to handle high call volumes or performance testing.

If you use automated testing software you **must ensure** that the volume is kept to the minimum required to validate that your application is working correctly and you must not submit the same call more times than is necessary to complete your tests.

You must review the results and conduct further testing, if required.  
**Once your testing is completed you must stop the automated testing tool and ensure that you are no longer connected to the site.**

**Claims Portal monitor the usage of the site and may contact you if they believe that your testing is not compliant with these A2A specification and instructions**



## Integration sites and testing

- **A2A QUERIES to be directed to the Helpdesk**
  - **helpdesk@rapidclaimsettlement.org.uk**
  - **with e-mail subject “A2A TOKEN QUERY”**
  - You should complete the A2A Integration/Test Contact Us form on the web site and ensure that you provide all information requested and attach to your email. Including the following information:
    - XML request
    - XML response including the Trace returned in the error message.
- **CHECK YOUR TEST CREDENTIALS**
  - please check that your TEST credentials are enabled a couple of weeks or so before you start testing. If you need support to enable the credentials, Don't leave to last minute as it may create delays.



## Q & A



**Thanks for your attention**



# Appendix





## Client ID's

CPL will be assigning each Software House with a unique Client ID.

The Software House will then add this ID to the 'Client ID' field as part of their A2A calls. This will illustrate the relationship between the Software House (via the 'Client ID' field) and their client (A2A user).

The 'Client ID' field already exists in the system and has a format of 'AANNNN'. As Claims Portal also want to be able to identify whether the Software House in question is an Internal Software House or an External Software House, the following approach in relation to the assignment of IDs to Software Houses will be followed:

- **Internal** Software Houses will be assigned an ID within the range of SH0001 - SH0999
- **External** Software Houses will be assigned an ID within the range of SH1000 - SH1999

### **Important:**

- An Internal Software House is defined as an organisation that has created a A2A solution internally for their own organisation's activities
- An external Software House is defined as a Software Provider that manages A2A messaging for a number of organization's