**Register Callback function**

By registering a callback the broker do not need to check / poll for updates on the credit decision, but a POST request will be sent to the registered URL.

Request

<setCallback>

    <callbackConfiguration>

            <callbackUrl>http://agent.com/url-to-callback</callbackUrl>

    </callbackConfiguration>

</setCallback>

Response

<setCallbackResponse>

    <setCallbackResult>

        <callbackSet>true</callbackSet>

        <message>Set callback successful</message>

    </setCallbackResult>

</setCallbackResponse>

**Comment**

Callbacks will be sent to the registered URL when application updated with following payload (content-type: application/json):

These events will trigger a notification:

CREDIT\_DECISION\_UPDATED, SIGNED, WAITING\_FOR\_ADDITION and PAID\_OUT

**Listning av de payloads som vi skickar:**

**SIGNED – when the signed agreement is received**

{

  "externalApplicationEvent" : {

    "applicationReference" : "ABC-123",

    "applicationEventType" : "SIGNED"

  },

  "jwsData" : "eyJOXAiOLADikz.eyJpc3MiJgbDQrm4.djjft4CP-mbW1"

}

**CREDIT\_DECISION\_UPDATED – APPROVED – when initial credit decision manual\_inspection is changed to approved**

{

  "externalApplicationEvent" : {

    "applicationReference" : "abc-123-def-567",

    "applicationEventType" : "CREDIT\_DECISION\_UPDATED",

    "creditDecision" : {

        "creditDecision" : "APPROVED",

        "approvedAmount" : 10000,

        "interest" : 12.3,

        "tenor" : 12,

        "effectiveInterest" : 23.3,

        "monthlyCost" : 1234,

        "consolidationDemand" : 2300,

        "adminFee" : 199,

        "arrangementFee" : 599.99,

        "totalRepaymentAmount" : 13000,

        "documentTypes" : [ "ID", "PAYMENT\_SLIP" ]

    }

  },

  "jwsData" : " eyJOXAiOLADikz.eyJpc3MiJgbDQrm4.djjft4CP-mbW1"

}

**CREDIT\_DECISION\_UPDATED - MANUAL\_INSPECTION**

{

  "externalApplicationEvent" : {

    "applicationReference" : "abc-123-def-567",

    "applicationEventType" : "CREDIT\_DECISION\_UPDATED",

    "creditDecision" : {

        "creditDecision" : "MANUAL\_INSPECTION",

        "approvedAmount" : 0

    }

  },

  "jwsData" : " eyJOXAiOLADikz.eyJpc3MiJgbDQrm4.djjft4CP-mbW1"

}

**PAID\_OUT – when the loan has been paid out to the customer**

{

  "externalApplicationEvent" : {

    "applicationReference" : "ABC-123",

    "applicationEventType" : "PAID\_OUT"

  },

  "jwsData" : " eyJOXAiOLADikz.eyJpc3MiJgbDQrm4.djjft4CP-mbW1"

}

**WAITING\_FOR\_ADDITION – when we are awaiting additional information from the customer**

{

  "externalApplicationEvent" : {

    "requiredDocuments" : [ "ID" ],

    "applicationReference" : "abc-123-def-567",

    "applicationEventType" : "WAITING\_FOR\_ADDITION"

  },

  "jwsData" : " eyJOXAiOLADikz.eyJpc3MiJgbDQrm4.djjft4CP-mbW1"

}

**jwsData**

The value in jwsData is a signed and serialized JWT, known as a JWS (JSON Web Signature). See RFC7515

JWS - A data structure cryptographically securing a JWS Header and a JWS Payload with a JWS Signature.



The payload consists of the JSON structure externalApplicationEvent. An example of a JWS payload could be

{

  "externalApplicationEvent" : {

    "applicationReference" : "ABC-123",

    "applicationEventType" : "PAID\_OUT"

  }

If the agents want to verify the JWS they should download our public keys from <https://apigw.resurs.com/api/auth_service/jwks>. Note that the customer must use an API key which is handed out by Resurs in order to call the URL above.

Example of layout of a public key

{

    "keys": [

        {

            "kty": "RSA",

            "e": "AQAB",

            "use": "sig",

            "kid": "1cb06a4a-3465-4f11-ac3d-d8bb2bdb1eb7",

            "alg": "RS256",

            "n": "lLtFi6baHG0yJfWKCFlfY5hMlqUoqvTWwFyjsKckuI62WVwv-GJ1vTPnCk9cptRjqGltQ4IJxv8FYOPPjaRi98li53BnaHRmg6pITF68OAIbnTpMn3\_hqctSS\_6XM2s-PFMfctbFuSj\_W4IzC2f1FmCXJSONz16zxy6kvlyC-ZVBLbN1WpWsheufGqR7tWibj780lgA8nwTQNHUXhdlydOU3CVfGEgs9IRx8vB53n0nTHwBdMI5i5ta9wEe2f7r6I4YqganJL\_q\_ilYfrbAB2yOCQz4AiWOpQzrhZCV54tiSGfz4lVfVS\_oJOy17LapdSqxnBmJwX\_R4uTR0g7BLWw"

        },

        {

            "kty": "RSA",

            "e": "AQAB",

            "use": "sig",

            "kid": "e434a27b-e1fb-4f7f-89d6-fb4e95619aac",

            "alg": "RS256",

            "n": "vUUYiYxqYwCHreCGlCBn0YjCBoLa-jrqfEhSLAWafCxaGFs5mAXrzb6Zzc\_cfzNptNQMtA4x2qhF68copSbGyXxq8ZQl-10J0cjLJ2HW98-5jJiGit3MIR5cEGEjMrRzdaltCFiGSIeussdJMOvNtJk7-OGMQgxMVMJ6PK5zARsNauj85WuB2UyOg6EDBRP7pfmQ9V4bBRGM-s3PQZbBCD-       Mcr1K0tQTc60wftRiC78AxlVubF60jpj6l1bcvpsdld8uF1wVIrU8gjh65CL9KTXMmb261PLrv-op8TSjFMT-DPw4Jb54RxqjbEO9ze6b6BJo-0eU7\_suqzitMT4JaQ"

        }

    ]

}