MANAGE QUALIFICATION IN A SINGLE SYSTEM

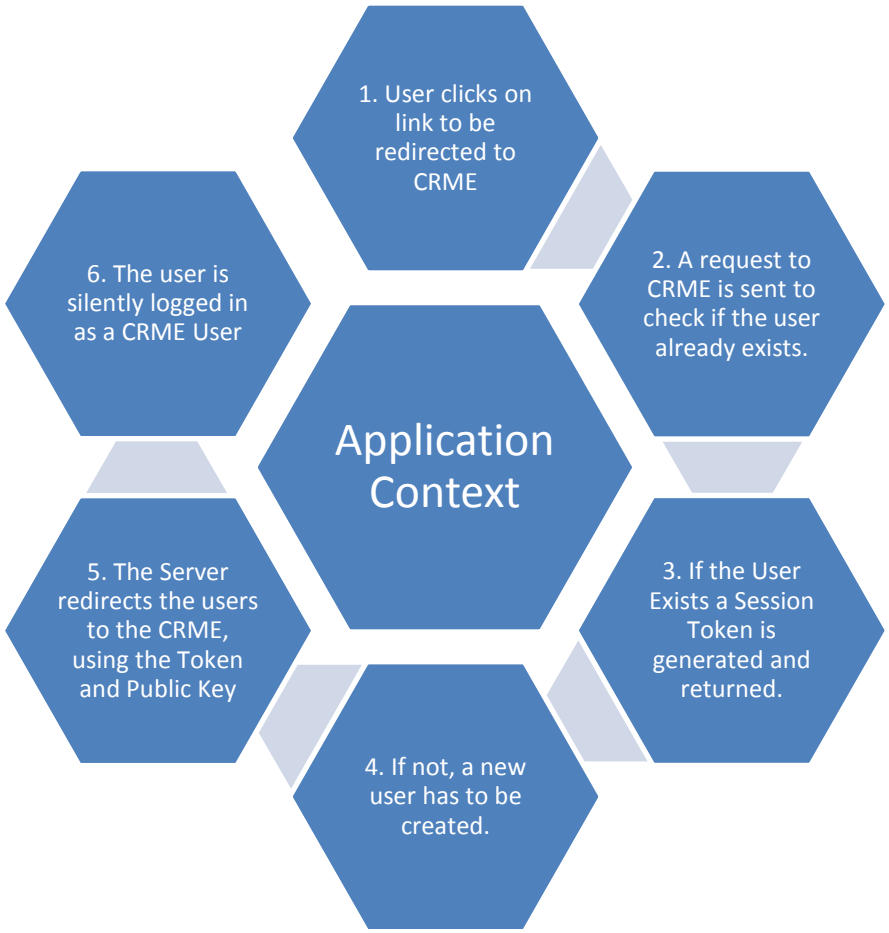
intelligent media systems ag


# CRME® 3.0

# SSO Integration Guide

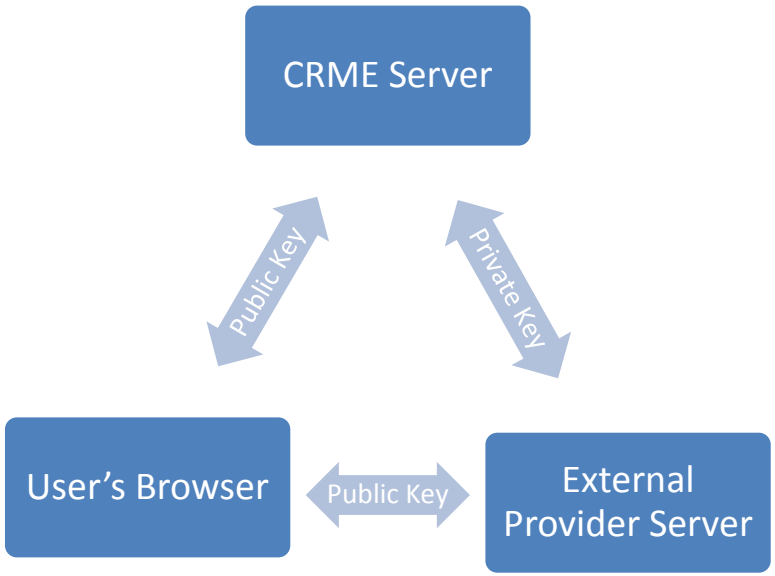
# Authentication Workflow

netTrainment 3.0



# Security Overview

netTrainment 3.0




- **Public Key**: a public key will be used for redirects.
- **Private Key**: a private key that should be used only for Server to Server communication.
- **IP Security**: Each Private API Key is associated with a set of IP Addresses and usage is allowed only from those specific IP Addresses
- Only **HTTPS encryption** is allowed

# Authenticating and Authorizing

netTrainment 3.0

intelligent media systems ag

- To **authenticate** the requests to CRME you have to set the Private Key as the Authorization Bearer token in each request.
- Example authentication header:

```
Authorization: Bearer mF_9.B5f-4.1JqM
```

- The Private Key could only be used for the API requests.

- /api/v1/auth
  - GET: checks for existing user by identifier
  - POST: create a new user
  - PUT: update an existing user

Usage Examples:

- GET: {url}/api/v1/auth/9nU2W01dJK
- PUT: {url}/api/v1/auth

{ “Identifier”: “9nU2W01dJK”, “FirstName”: “John”, “FastName”: “Doe”, “Email”: “john@doe.org” … }

- If the request fails, the response will have a status of 400 Bad Request and will have contents as follows:

| Parameter | Description |
|---|---|
| BODY DATA | A JSON object with the following properties: |
| | error: (string) An error code. |
| | error_description: (string) A more detailed description of the error intended for the developer of your app. |

- For example, you might receive the following data in the response body after a failed request:

```
{ "error" : "invalid_request", "error_description" : "One or more parameters are missing: client_secret" }
```

The following JSON object will be used for the API requests:

| Name | Datatype | | Description/Restrictions |
|---|---|---|---|
| Identifier | String | X | Unique Identifier to identify the user account in the external provider. Max 256 chars |
| UserName | String | X | Max 256 chars |
| Email | String | X | Max 256 chars |
| IsNonUniqueEmail | Bool | X | Boolean with true / false (default is false) |
| FirstName | String | X | Max 100 chars |
| LastName | String | X | Max 100 chars |
| CountryCode | String | X | Country two letter ISO Code e.g. GB |
| LanguageCode | String | X | Region ISO Code e.g. en-GB |
| ActivationCode | String | | Max 200 chars |
| AuthorizationToken | String | | Max 256 chars **(will be provided by CRME)** |
| Expiration | Timestamp | | e.g. 1448046245 **(will be provided by CRME)** |

***All required fields are marked with an X within the table above.***
***All chars are unicode.***

# GET /api/{version}/auth

netTrainment 3.0

intelligent media systems ag

| | |
|---|---|
| **Method/Request** | **GET** /api/v1/auth/{id} |
| **Summary** | Get existing user by identifier |
| **Parameters** | |

| Name | Type | Located In | Description | Required |
|---|---|---|---|---|
| Identifier | String | URL | Unique Identifier of the User | Yes |

**Response**

| Http Response Code | Description |
|---|---|
| 200 | Successful, returns a User Model object with the authorization token and expiration timestamp |
| 400 | Invalid data |
| 404 | User not found |

# PUT /api/{version}/auth/{id}

netTrainment 3.0

intelligent media systems ag

| | |
|---|---|
| **Method/Request** | **POST** /api/v1/auth/{id} |
| **Summary** | Create a new user User with identifier {id} |
| **Parameters** | |

| Name | Type | Located In | Description | Required |
|---|---|---|---|---|
| User | User Model | Body | The user model to be created | Yes |

**Response**

| Http Response Code | Description |
|---|---|
| 200 | Successful, returns a User Model object |
| 400 | Invalid data |

# POST /api/{version}/auth/{id}

netTrainment 3.0

intelligent media systems ag

| Method/Request | **PUT** /api/v1/auth/{id} |
|---|---|
| Summary | Update an existing User with identifier {id} |

**Parameters**

| Name | Type | Located In | Description | Required |
|---|---|---|---|---|
| Identifier | String | URL | Unique Identifier of the User | Yes |
| User | User Model | Body | The user model with the updated properties | Yes |

**Response**

| Http Response Code | Description |
|---|---|
| 200 | Successful, returns a User Model object |
| 400 | Invalid data |

- The External Provider requests to sign in a user to the CRME by using the **GET request**.
  - In case the requested user does not exist, a new user is created by using the **PUT request**.

- If the request is successful, the response will deliver a **user model** with the **Authorization Token** and **Expiration Timestamp** for this user.

- The user can be then **redirected** to **CRME** by using the **Public Key and Authorization Token**.

# Redirection to CRME via [GET]

netTrainment 3.0

- After the **AuthorizationToken** has been delivered to the External Provider this is used to sign in the user to the CRME via GET:
  - **Redirection URL**: https://crme.com/api/oauth2/Authenticate?PublicKey={PublicKey}&Token={AuthorizationToken} &ReturnUrl={*URL*}

- The request required data:
  - **Public Key** is used to identify the External Provider
  - **AuthorizationToken** of the specific user
  - **ReturnUrl** to the target website

- This will work **only** if the Authorization Token has **not expired**.

- If the token has expired, a redirection back to the External Authentication Provider will occur and a failed status result will be returned. It is recommended to store the **AuthorizationToken** and **Expiration** in order to refresh it when necessary.

- To refresh an **AuthorizationToken** another **GET request** has to be made and will be then provided within the **User Model**.

# Redirection to CRME via [POST]

netTrainment 3.0

- After the **AuthorizationToken** has been delivered to the External Provider this is used to sign in the user to the CRME via POST:
  - **Redirection URL**: https://crme.com/api/oauth2/Authenticate

- The request required data:
  - **Public Key** is used to identify the External Provider
  - **AuthorizationToken** of the specific user
  - **ReturnUrl** to the target website

- This will work **only** if the Authorization Token has **not expired**.

- Sample script to submit a redirection request via POST:

```
function jspost(path, params, method) {
        method = method || "post";
        var form = document.createElement("form");
        form.setAttribute("method", method);
        form.setAttribute("action", path);
        for (var key in params) {
          if (params.hasOwnProperty(key)) {
            var hiddenField = document.createElement("input");
            hiddenField.setAttribute("type", "hidden");
            hiddenField.setAttribute("name", key);
            hiddenField.setAttribute("value", params[key]);
            form.appendChild(hiddenField);
          }
        }
        document.body.appendChild(form);
        form.submit();
    }
```