**Search4**

**API Documentation**

**USERS**

* **Return** **all** currently active users on Search4:

HTTP Method: GET

URL: {Base API URL}/users

Example: http://localhost:8080/Search4/api/v1/users

Returns: JSON List of users

* **Return** **one** user from Search4 based on **id** number:

HTTP Method: GET

URL: {Base API URL}/users/{id}

Example: http://localhost:8080/Search4/api/v1/users/1

Returns: JSON user

* Add a **new** **user** to Search4:

HTTP Method: POST

URL: {Base API URL}/users

Example: http://localhost:8080/Search4/api/v1/users

HEADER INFO: Content-Type = application/json

JSON BODY: {

"firstName" : "John",

"lastName" : "Doe",

"email" : "johndoe@alive.com",

"password" : "password4Me"

}

Limitations: **firstName** must be less than or equal to 255 characters

**lastName** **not required**

**email** must be a new (unique) email adress and less than or equal to 255 characters

**password** must be 4 or more characters

Returns: JSON user

* **Edit/update** an existing **user** registered to Search4:

this API can only update a user's first name, last name and/or email address and requires valid login information (old email and correct password)

HTTP Method: PUT

URL: {Base API URL}/users/{id}

Example: http://localhost:8080/Search4/api/v1/users/1

HEADER INFO: Content-Type = application/json

JSON BODY: {

"email" : "johndoe@alive.com",

"password" : "password4Me",

"firstName" : "Johnny",

"lastName" : "Doeson",

"updatedEmail" : "johnnydoeson@gmail.com"

}

Limitations: **email** must be the old email address

**password** must be the correct password

**firstName** **not** **required** - must be less than or equal to 255 characters.

**lastName** **not** **required**

**updatedEmail** **not** **required** - must be must be a new (unique) email adress and less than or equal to 255 characters

Returns: Transaction Information

* **Delete** **one** user from Search4 based on **id** number:

HTTP Method: DELETE

URL: {Base API URL}/users/{id}

Example: http://localhost:8080/Search4/api/v1/users/1

Returns: Transaction Information

* **Return** **all providers** saved for one user from Search4 based on **user** **id** number:

HTTP Method: GET

URL: {Base API URL}/users/{id}/providers

Example: http://localhost:8080/Search4/api/v1/users/1/providers

Returns: JSON List of providers

* **Return** **all subscriptions** saved for one user from Search4 based on **user** **id** number:

HTTP Method: GET

URL: {Base API URL}/users/{id}/subscriptions

Example: http://localhost:8080/Search4/api/v1/users/1/subscriptions

Returns: JSON List of subscriptions

**Providers**

* **Return all** providers from Search4:

HTTP Method: GET

URL: {Base API URL}/providers

Example: <http://localhost:8080/Search4/api/v1/providers>

Returns: JSON list of providers

* **Return one** provider by provider id from Search4:

HTTP Method: GET

URL: {Base API URL}/providers/providerid/{id}

Example: <http://localhost:8080/Search4/api/v1/providers>/providerid/1

Returns: JSON provider

* **Return one** provider by user id from Search4:

HTTP Method: GET

URL: {Base API URL}/providers/userid/{id}

Example: <http://localhost:8080/Search4/api/v1/providers>/userid/1

Returns: JSON provider

* **Add a new provider** with userid to Search4:

HTTP Method: POST

URL: {Base API URL}/providers/{id}

Example: <http://localhost:8080/Search4/api/v1/providers/1>

HEADER INFO: Content-Type = application/json

JSON BODY:

{

”name”: ”providerName”

}

Returns: JSON provider

* **Delete a provider** with provider id from Search4:

HTTP Method: DELETE

URL: {Base API URL}/providers/{id}

Example: <http://localhost:8080/Search4/api/v1/providers/1>

HEADER INFO: Content-Type = application/json

JSON BODY:

{

”name”: ”providerName”

}

Returns: Transaction information