[User roles table](https://docs.google.com/spreadsheets/u/3/d/1wHPDfGIJjxQoZR0tun432C2QtJbNpFmWuPXgzhOZieI/edit?usp=drive_web&ouid=114159974584356895936)

[1. Main concept:](#_va9fg9evxu82)

[1.1. Format: JSON](#_zfjs4cswui53)

[1.2. Request](#_zfjs4cswui53)

[1.3. Answer](#_99cnt96oo8qu)

[1.4 Generic errors](#_wgfplf5aufop)

[2. Get config](#_pzh7eptqmt0q)

[3. Login](#_br1j7vb3xuwx)

[3.1 Restore session](#_ef0kkhqdifth)

[3.2. Ping user](#_g0lmrfikzpe5)

[4. Password recovery](#_tawfsai264q4)

[5. Password recovery - change password](#_k6i42qck85te)

[6. Enterprise registration](#_45a8bi1rgggg)

[7. Enterprise registration - email confirmation](#_os5c0vfff5kj)

[8. Add/edit Enterprise](#_yzak3ht2g09r)

[9. Get Enterprises data](#_6jme42956mum)

[10. Transfer Enterprise ownership](#_g0h4xio87fv9)

[10.1 Grant Enterprise ownership](#_dbsgmhotknxj)

[10.2 Revoke Enterprise ownership](#_m8bsz3ymm0gd)

[10.3 Grant/Revoke Enterprise ownership](#_hyk1wx0whpg)

[11. Factories - get data](#_kfcsagp70fdk)

[12. Factory add/edit](#_12z8iaesewh)

[12.1 Grant Factory ownership](#_3xzyci9g8c12)

[12.2 Revoke Factory ownership](#_b6jnbihyxvo)

[12.3 Grant/Revoke Factory ownership](#_ldjfa7gj5px)

[13. Add monthly raw Material Input - Config Get](#_6l0fc73kqklb)

[14.1. Add monthly raw Material Input](#_f4arycdj2tkb)

[14.3. View monthly raw Material Input](#_g000n9j2121y)

[15. Add monthly Production Data - Config Get](#_q2kq3brm9811)

[15.1. Add monthly Production Data](#_766fo5ptpmrb)

[15.2. View monthly raw Production Data](#_hhmhvri938xj)

[15.3. Remove monthly raw Production Data](#_71svribmzppy)

[16. Get accounts data](#_16ecivca764t)

[17. Add/edit account](#_cbdkwu23z1kt)

[18. Delete account](#_pi9zc9e4bczz)

[19. Delete factory](#_p7lftpnqtddp)

[20. Delete enterprise](#_v3xeqk7gh5wk)

[21. Save map settings](#_3ir38repyotx)

[22. Load map settings](#_4mp9p42056e5)

[23. Search factories (for reports)](#_82a06kitk7kw)

[24. Search factories for monitoring report](#_ojb31nu8haww)

[25. Get quick report in excel file](#_h9lo1755vuu4)

[26. Get monitoring report in excel file](#_4s0vqksm9ail)

[27. Set reminder settings](#_2ydnl1eo5i0b)

[28. Get reminder settings](#_ei99vb2v4l41)

[29. Inspection report upload](#_a2m27ygm0a8o)

[30. Inspection report download](#_ruhorwkur45x)

[31. Inspection report listing](#_x1534yulrs1a)

[32. Fetching admin area bounds](#_x5k653sq81g2)

## 1. Main concept:

### 1.1. Format: JSON

### 1.2. Request

{

sid: "....", // only if we're logged in

role: “...”, // only if we’re logged in; defines active role

func1: { // procedure name

execOrder: 1, // optional, execution order if it's important

func1param: "func1value", ...

func1param2: {

func1value2: "..."

},

...

},

func2: {

func2param: "...",

...

},

}

### 1.3. Answer

{

func1: { // same procedure name

func1results: func1resultValues, ...

},

func2: {

errorCode: "loginWrongPassword", // we will search this in language file

error: "Error text", // show this if we can't find proper error translation in language file

}

}

The rest of this document lists the supported procedures (func1, func2, ... in 1.2 & 1.3) and their parameters (func1params, func2params... in 1.2 & 1.3)

1.4. All ids are hash-strings, because we don't want to show the total number of items

1.5. All the queries require sid except configGet, login, passwordRecovery, passwordRecoveryChange, enterpriseRegister

### 1.4 Generic errors

There are two generic errors the system can produce:

First, in the case that the input is malformed (not valid JSON) (http code 400)

{

"errorCode": "inputBad",

"error": "Invalid input format",

"report": string, // only if exists

}

Second, a server error that is not otherwise caught (http code 500)

{

"errorCode": "uncaughtError",

"error": "Uncaught error (report uid …)",

"report": string, // only if exists

}

## 2. Get config

Note: params are in <https://qa.simosol.fi/issues/4664> and <https://qa.simosol.fi/issues/4663>

Request:

{ configGet: true }

Answer:

{ configGet:

{

classByOwner: { key: val, ... }, // for enterprises and factories

classByCapitalOrigin: { key: val, ... }, // for enterprises

classByMainProducts: { key: val, ... }, // for enterprises

reportTypes: { key: val, ... }, // for reports

industryType: { key: val, ... }, // for factories

industryGroup: { key: val, ... }, // for factories

certificate: { key: val, ... }, // for factories

ocsClass: { key: val, ... }, // for factories

unit: { key: val, ... }, // for factories, m2, ton

productType: { key: val, ... },

province: [ {code: 'code', name: 'name' },... ],

district: [ {code: 'code', name: 'name', province: province\_code},... ]

commune: [ {code: 'code', name:'name', province: province\_code,

district: district\_code }, ... ],

userTypes: { typeCode: typeName, ... } // all possible user roles

}

}

## 3. Login

Note: permissions currently not included  
Note: Q-REQ 15: If a user enters their password incorrectly a given number of times in a row in one session, their account shall be blocked. The tolerable number of times shall be configurable per user class.

Request:

{ login: { email: ".@..",

password: "...",

type: "..." } } // optional - defines active role if role is not in later calls

Answer:

{ login:

{

sid: "sid hash",

id: "hash",

username: "...",

email: "...",

primary\_type: "DistrictManager", // check <https://qa.simosol.fi/issues/4657>

secondary\_type: "Enterprise", // optional, only for valid primary\_types

lastLoginDate: "2018.02.04",

lastLoginIp: "87.250.250.242",

region: "code", // optional, for regional, province and district manager

province: "code", // optional, for province and district manager

district: "code", // optional, for district manager

bounds: [<minx>, <miny>, <maxx>, <maxy>], // optional, for regional, province and district manager, for enterprise and factory around owned factories if any

permissions: {

enterpriseAdd: true,

enterpriseEdit: true,

...

}

}

}

Error:  
{ errorCode: "loginBad", error: "Invalid username or password" }

{ errorCdoe: "accountDisabled", error: "The given account has been disabled"}

## 3.1 Restore session

Note: permissions currently not included

Request:

{ restoreSession: { sid: "sid hash" } }

Answer:

{ restoreSession:

{

sid: "sid hash",

id: "hash",

username: "...",

email: "...",

primary\_type: "DistrictManager", // check <https://qa.simosol.fi/issues/4657>

secondary\_type: "Enterprise", // optional, only for valid primary\_types

lastLoginDate: "2018.02.04",

lastLoginIp: "87.250.250.242",

region: "code", // optional, for regional, province and district manager

province: "code", // optional, for province and district manager

district: "code", // optional, for district manager

bounds: [<minx>, <miny>, <maxx>, <maxy>], // optional, for regional, province and district manager, for enterprise and factory around owned factories if any

permissions: {

enterpriseAdd: true,

enterpriseEdit: true,

...

}

}

}

Error:

{errorCode: "invalidSession", error: "Invalid or expired session"}

## 3.2. Ping user

Note: Q-REQ 13: The application shall support user inactivity time-out.

Request:

{ pingSession: { sid: "sid hash" } }

Answer:

{ pingSession: { done: true } }

Error:

{ errorCode: "invalidSession", error: "Invalid or expired session" }

## 4. Password recovery

Request:

{ passwordRecovery: { email: ".@.." } }

Answer:

{ passwordRecovery: { done: true } }

Errors:

{errorCode: "accountDisabled", error: "The given account has been disabled}

{errorCode: "accountDoesntExist", error: "Given email is not registered in the system"}

## 5. Password recovery - change password

URL: /#recoveryChange/<recoveryToken>

! We need to define the rules for safe passwords and check them on front- and back-end.

Request:

{ passwordRecoveryChange: { recoveryToken: "...", passwordNew: "..." } }

Answer:

{ passwordRecoveryChange: {

done: true,

} }

Errors:

{errorCode: "invalidConfirmationCode", error: "The given confirmation code is invalid or outdated"}

{errorCode: "accountDoesntExist", error: "Given email is not registered in the system"}

{errorcode: "accountDisabled", error: "The given account has been disabled}

## 6. Enterprise registration

Request:

{ enterpriseRegister:

{

account: {

email: ".@..",

password: "...",

username: "...", // not sure about this field

telephone: "...",

},

enterprise: {

name: "...",

taxCodeNumber: "...", // unique

taxCodeRegistrationDate: "yyyy.mm.dd", // back-end should check for valid date

address: "...",

email: ".@..",

website: “...”, // optional

telephone: "...",

owner: "...",

affiliatedUnits: "...",

classByOwner: "...", // from the set config.classByOwner

classByCapitalOrigin: "...", // from the set config.classByCapitalOrigin

classByMainProducts: ["...", …], // from the set config.classByMainProducts

forestNaturalArea: 123.5,

forestPlantationArea: 123.5,

forestRubberArea: 123.5,

forestScatteredArea: 123.5

},

recaptchaResponse: token // read this https://developers.google.com/recaptcha/docs/verify

}

}

Answer:

{ enterpriseRegister: { done: true, email: "..." } }

Errors:

{"errorCode": "accountAlreadyExists", "error": "Given email is already registered into the system"}

{"errorCode": "enterpriseAlreadyExists", "error": "Enterprise with the given name is already registered into the system"}

## 7. Enterprise registration - email confirmation

Request:

{ enterpriseRegisterConfirm: { regid: "hash" } }

Answer:

{ enterpriseRegisterConfirm: {

done: true,

sid: "sid hash",

id: "hash",

username: "...",

email: "...",

primary\_type: "Enterprise", // check <https://qa.simosol.fi/issues/4657>

lastLoginDate: "2018.02.04",

lastLoginIp: "87.250.250.242",

permissions: {

enterpriseAdd: true,

enterpriseEdit: true,

...

}

}

Errors:

{"errorCode": "invalidConfirmationCode", "error": "The given confirmation code is invalid or outdated"}

{"errorCode": "accountAlreadyExists", "error": "Given email is already registered into the system"}

{"errorCode": "enterpriseAlreadyExists", "error": "Enterprise with the given name is already registered into the system"}

## 8. Add/edit Enterprise

NOTE: Only Enterprise users can add new enterprises; Only Enterprise users can edit, and only their owned enterprises

Request:

{ enterpriseAdd: / enterpriseEdit:

{

id: "hash", // only for editing, unique id for enterprise

name: "...", // unique

taxCodeNumber: "...",

taxCodeRegistrationDate: "yyyy.mm.dd", // back-end should check for valide date

address: "...",

email: ".@..",

website: “...”, //optional

telephone: "...",

owner: "...",

affiliatedUnits: "...",

classByOwner: "...", // value from the set from config

classByCapitalOrigin: "...", // value from the set from config

classByMainProducts: ["...", …], // value from the set from config

forestNaturalArea: 123.5,

forestPlantationArea: 123.5,

forestRubberArea: 123.5,

forestScatteredArea: 123.5

}

}

Answer:

{ enterpriseAdd / enterpriseEdit: { done: true, id: "hash" } }

Errors (add):

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseAlreadyExists, error: Enterprise with the given name is already registered into the system}

Errors (edit):

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

{errorCode: enterpriseAlreadyExists, error: Enterprise with the given name is already registered into the system}

## 9. Get Enterprises data

NOTE: Enterprise and Factory users only see their associated enterprises. Everyone else sees all enterprises.

Request:

{ enterpriseDataGet:

{

id: "...", // optional

search: "...", // optional, non-restricted search by name

short: true / false, // optional, return only id, name and factoriesAmount (for list)

ownedOnly: boolean, // optional, with the flag factory users no longer see their parents

}

}

Answer:

{ enterpriseDataGet:

{

items: [

{

id: "hash",

name: "...",

taxCodeNumber: "...",

taxCodeRegistrationDate: "yyyy.mm.dd", // back-end should check for valide date

address: "...",

email: ".@..",

website: “...”,

telephone: "...",

owner: "...",

affiliatedUnits: "...",

classByOwner: "...", // value from the set

classByCapitalOrigin: "...", // value from the set

classByMainProducts: ["...", ...], // values from the set

forestNaturalArea: 123.5,

forestPlantationArea: 123.5,

forestRubberArea: 123.5,

forestScatteredArea: 123.5,

factoriesAmount: 123,

},

{ id: "...", name: "...", factoriesAmount: 0 }, // if short === true

...

]

}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 10. Transfer Enterprise ownership

Note: this will remove the enterprise from the user making the call); Only for owned enterprises

Request:

{ enterpriseTransfer: { email: ".@..", enterpriseId: "..." } }

Answer:

{ enterpriseTransfer: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: enterpriseNotOwned, error: You do not have control over the given enterprise}

{errorCode: enterpriseAlreadyOwned, error: The recipient already controls this enterprise}

## 10.1 Grant Enterprise ownership

NOTE: Owned, unless national admin

Request:

{ enterpriseGrant: { email: ".@..", enterpriseId: "..." } }

Answer:

{ enterpriseGrant: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: enterpriseNotOwned, error: You do not have control over the given enterprise}

## 10.2 Revoke Enterprise ownership

NOTE: Owned, unless national admin

Request:

{ enterpriseRevoke: { email: ".@..", enterpriseId: "..." } }

Answer:

{ enterpriseRevoke: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: enterpriseNotOwned, error: You do not have control over the given enterprise}

## 10.3 Grant/Revoke Enterprise ownership

NOTE: Owned, unless national admin

Request:

{ enterpriseGrantRevoke: { email: ".@..", enterpriseIds: {“eid”: boolean} } }

// true grant, false revoke

Answer:

{ enterpriseGrantRevoke: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: enterpriseNotOwned, error: You do not have control over the given enterprise}

## 11. Factories - get data

NOTE:

* SHORT data is available from all factories for all users
* Otherwise only factory and enterprise users can get results and only for factories they control

Request:

{ factoryDataGet:

{

id: "...", // optional

enterpriseId: "...", // optional

search: "...", // optional, non-restricted search by name or number or telephone or owner or address

short: true / false, // optional, return only id, name, number, telephone, owner and address (for list) + website, industryGroup, productType, certificate, certificateOther, designedInputCapacity, designedInputCapacityUnit, designedOutputCapacity, designedOutputCapacityUnit

}

}

Answer:

{ factoryDataGet:

{

items: [

{

id: "hash",

enterpriseId: “hash”,

name: "...",

enterpriseName: “...”,

taxCodeNumber: "...",

address: "...",

telephone: "...",

email: ".@..",

owner: bool,

affiliatedUnits: "...",

coordinateX: 12.1,

coordinateY: 34.5,

province: "...",

district: "...",

commune: "...",

industryType: "...", // from the set config.industryType

industryGroup: "...", // from the set config.industryGroup

classByOwner: "...", // from the set config.classByOwner

certificate: ["...", "..."], // from the set config.certificate, array

certificateOther: “...”, // IFF certificate “Other” is in the list

ocsClasses: "...", // from the set config.ocsClasses

designedInputCapacity: 12.3,

designedInputCapacityUnit: "...", // from the set config.unit

designedOutputCapacity: 18.2,

designedOutputCapacityUnit: "...", // from the set config.unit

management: 12,

supervision: 34,

laborers: 2,

supportStaff: 35,

otherStaff: 6,

productType: "...", // from the set config.productType

byProductTypes: [“...”, ...], // additional productTypes

website: string; // from enterprise data

},

]

}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 12. Factory add/edit

NOTE: Adding only for enterprise users, editing also for factory users (for both, only owned)

Request:

{ factoryAdd: / factoryEdit:

{

id: "hash", // only for editing, unique id for factory

enterpriseId: "hash",

name: "...",

taxCodeNumber: "...",

address: "...",

telephone: "...", // optional

email: ".@..",

owner: "...",

affiliatedUnits: "...", // optional

coordinateX: 12.1,

coordinateY: 34.5,

provinceName: "...",

districtName: "...",

communeName: "...",

industryType: "...", // from the set config.industryType

industryGroup: "...", // from the set config.industryGroup

classByOwner: "...", // from the set config.classByOwner

certificate: ["...", "..."], // from the set config.certificate, array

certificateOther: “...”, // IFF certificate “Other” is in the list

ocsClasses: "...", // from the set config.ocsClasses

designedInputCapacity: 12.3,

designedInputCapacityUnit: "...", // from the set config.unit

designedOutputCapacity: 18.2,

designedOutputCapacityUnit: "...", // from the set config.unit

management: 12,

supervision: 34,

laborers: 2,

supportStaff: 35,

otherStaff: 6,

productType: "...", // Sawn timber and Wood chips only for **industryType === 1** // from config.productType

byProducTypes: [“...”, ...] // additional productTypes

}

}

Answer:

{ factoryAdd / factoryEdit: { done: true, id: "hash" } }

Errors (add and edit):

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

Errors (edit):

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

## 12.1 Grant Factory ownership

NOTE: Owned, unless national admin

Request:

{ factoryGrant: { email: ".@..", factoryId: "..." } }

Answer:

{ factoryGrant: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: factoryNotOwned, error: You do not have control over the given factory}

## 12.2 Revoke Factory ownership

NOTE: Owned, unless national admin

Request:

{ factoryRevoke: { email: ".@..", factoryId: "..." } }

Answer:

{ factoryRevoke: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: factoryNotOwned, error: You do not have control over the given factory}

## 12.3 Grant/Revoke Factory ownership

NOTE: Owned, unless national admin

Request:

{ factoryGrantRevoke: { email: ".@..", factoryIds: {“fid”: <boolean> } }

// true grants, false revokes

Answer:

{ factoryGrantRevoke: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

{errorCode: accountDoesntExist, error: Given email is not registered in the system}

{errorCode: factoryNotOwned, error: You do not have control over the given factory}

## 13. Add monthly raw Material Input - Config Get

Request:

{ factoryMonthlyMaterialInputConfigGet: true }

Answer:

{ factoryMonthlyMaterialInputConfigGet:

{

“rawMaterialSource”: {“<code>”: “<description>”},

“exploitationAgency”: {“<code>”: “<description>”},

“rawMaterialCertificationProgram”: {“<code>”: “<description>”},

“forestOwnerType”: {“<code>”: “<description>”},

“volumeByClasses”: {“<code>”: “<description>”} // This is for all volumeByClasses\* and

// storageVolumeByClasses\* parameters,

// contains tree species.

}

}

## 14.1. Add monthly raw Material Input

+ <https://qa.simosol.fi/issues/4663>

<https://qa.simosol.fi/issues/4664>

Note: uses codes as keys, not names

Note: it rewrites the old one with the same month and year

NOTE: Only by Enterprise/Factory user that is owner of the factory

Request:

{ factoryMonthlyMaterialInputAdd: // All fields are optionsl (except factory id), but at least one of them must exists

{

factoryId: "hash",

month: 2,

year: 2018,

// All volumeByClasses have the same syntax as volumeByClassesOther

// Codes are from volumeByClasses

volumeByClassesRoundwood: [

volumeByClassesSawnTimber: [

volumeByClassesPanels: [

volumeByClassesPlywood: [

volumeByClassesPulp: [

volumeByClassesWoodChips: [

volumeByClassesBiofuel: [

volumeByClassesOther: [

{ type: "Code 2", amount: 800, unit: "1" }, // from the set

...

],

// All storageVolumeByClasses have the same syntax as storageVolumeByClassesOther

// Codes are from volumeByClasses

storageVolumeByClassesRoundwood: [

storageVolumeByClassesSawnTimber: [

storageVolumeByClassesPanels: [

storageVolumeByClassesPlywood: [

storageVolumeByClassesPulp: [

storageVolumeByClassesWoodChips: [

storageVolumeByClassesBiofuel: [

storageVolumeByClassesOther: [

{

{ type: "Code 2", amount: 800, unit: "1" }, // from the set

...

],

forestOwnerType: [ // only for **industryType === ‘1’**

{ type: "...", amount: 100, unit: "1" }, // from the set

...

],

rawMaterialSource:[ // only for **industryType === ‘1’**

{ type: "...", amount: 100, unit: "1" }, // from the set

...

],

exploitationAgency:[ // only for **industryType === ‘1’**

{ type: "...", amount: 100, unit: "1" }, // from the set

...

],

rawMaterialCertificationProgram:[

{ type: "...", amount: 100, unit: "1", addInfo: “12345” }, // from the set; addInfo for license number

...

],

rawMaterialOrigin:[

{ type: "...", amount: 100, unit: "1" }, // from the set

...

],

HSStandard:[

{ type: "...", amount: 100, unit: "1" }, // 6 digit code as type

...

],

}

}

Answer:

{ factoryMonthlyMaterialInputAdd: { done: true, id: "hash" } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

## 14.3. View monthly raw Material Input

Request:

{factoryMonthlyMaterialInputGet:

{

id: hash // selection - input uid, optional

month: 2, year: 2018, // Search by month, year, optional

factoryId: hash // selection - factory uid

short: true // optional - short form has uid, year, month and byUser

}

}

Answer:

{factoryMonthlyMaterialInputGet:

{input: [

{

uid: hash,

year: 2018,

month: 2,

byUser: username

values: { ...check 14.1. Add monthly raw Material input }

}

]

}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 15. Add monthly Production Data - Config Get

Request:

{ factoryMonthlyProductionConfigGet: true }

Answer:

{ factoryMonthlyProductionConfigGet:

{

**storageVolume: {“<code>”: “<description>”},**

**productCategory: {“<code>”: “<description>”},**

**targetMarket: {“<code>”: “<description>”},**

**certificationProgram: {“<code>”: “<description>”},**

“volumeByClasses”: {“<code>”: “<description>”} // This is for productCategory and

// storageVolume subclasses,

// contains tree species.

}

}

## 15.1. Add monthly Production Data

+ https://qa.simosol.fi/issues/4663

Note: uses codes as keys, not names

Note: it will rewrite the old one with the same month and year

NOTE: Only by Enterprise/Factory user that is owner of the factory

Request:

{ factoryMonthlyProductionAdd: // All fields are optionsl (except factory id), but at least one of them must exists

{

factoryId: "hash",

month: 2,

year: 2018,

productCategory:

[{type: “...”,

items: [

{ type: "...", amount: 100, unit: "1" }], // from the set

…]

],

certificationProgram:[{type: “...”,

items: [

{ type: "...", amount: 100, unit: "1", addInfo: “12345” }, // from the set; addInfo for license

…]

...

],

targetMarket:[{type: “...”,

items: [

{ type: "...", amount: 100, unit: "1" }, // from the set

…]

...

],

storageVolume:[{type: “...”, items: [

{ type: "...", amount: 100, unit: "1" }], // from the set

...

],

HSStandard:[

{ type: "...", amount: 100, unit: "1" }, // 6 digit code as type

...

],

}

}

Answer:

{ factoryMonthlyProductionAdd: { done: true, id: "hash" } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

## 15.2. View monthly raw Production Data

Request:

{factoryMonthlyProductionGet:

{

id: hash // selection - production uid, optional

month: 2, year: 2018, // Search by month, year, optional

factoryId: hash // selection - factory uid

short: true // optional - short form has uid, year, month and byUser

}

}

Answer:

{factoryMonthlyProductionGet:

{

productions: [

{

id: hash,

year: 2018,

month: 2,

byUser: username

values: { ...check 15.1. Add monthly raw Production input }

}

]

}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: productDoesntExist, error: The given production entry does not exist in the system}

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

## ~~15.3. Remove monthly raw Production Data~~

Please remove this for now, looks like we don’t need this

Request:

{factoryMonthlyProductionRemove:

{id: hash} // production uid

}

Answer:

{factoryMonthlyProductionRemove:

{done: true}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: productDoesntExist, error: The given production entry does not exist in the system}

## 16. Get accounts data

NOTE:

* Managers can only get data on themselves
* Enterprise and factory users can only get data from associated enterprises
* Provincial admins can only get data from managers in their province and all enterprise and factory users
* National admins can get data from all users

Request:

{ accountsDataGet:

{

id: "...", // optional

search: "...", // optional, non-restricted search by name and enterprise name

short: true / false // optional, return only id, name, enterprise, group name, type, status

}

}

Answer:

{ accountsDataGet:

{

items: [

{

id: "hash",

primary\_type: "DistrictManager", // check <https://qa.simosol.fi/issues/4657>

secondary\_type: "Enterprise", // optional, only for valid primary\_types

status: "enabled", // value from the set [enabled, disabled, emailNeedConfirmation]

email: ".@..",

username: "...", // not sure about this fiel

telephone: "...",

enterprises: [id1, id2, ...], // optional, for enterprise and factory managers

factories: [id, ...], // optional, for enterprise and factory managers

region: "code", // optional, for regional, province and district manager

province: "code", // optional, for province and district manager

district: "code" // optional, for district manager

}

}

}

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 17. Add/edit account

NOTE: Only admins and enterprise users can add new users

* Enterprise users can add only enterprise- and factory users
* Provincial admins can add only provincial managers and district managers (inside their province)
* For editing: see <https://docs.google.com/spreadsheets/d/1wHPDfGIJjxQoZR0tun432C2QtJbNpFmWuPXgzhOZieI/edit#gid=1965868446>

Request:

{ accountAdd: / accountEdit:

{

id: "hash", // only for editing, unique id for account

primary\_type: "DistrictManager", // check <https://qa.simosol.fi/issues/4657>

secondary\_type: "Enterprise", // optional, only for valid primary\_types

status: "enabled", // value from the set [enabled, disabled]

email: ".@..",

username: "...", // not sure about this field

password: "...", // optional, change the password if exists

telephone: "...",

enterprises: [id1, id2, ... ], // optional, for enterprise managers ONLY on add

factories: [id1, id2, ... ], // optional, for factory managers ONLY on add

region: "code", // optional, for regional manager

province: "code", // optional, for province manager

district: "code", // optional, for district manager

// setup input reminder email sending

reminderDay: <day-of-month>, // optional (reminder removed if missing)

reminderHour: <hour-of-day> // optional (reminder removed if missing)

}

}

Answer:

{ enterpriseAdd / enterpriseEdit: { done: true, id: "hash" } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

{errorCode: factoryDoesntExist, error: The given factory does not exist in the system}

{errorCode: enterpriseDoesntExist, error: The given enterprise does not exist in the system}

## 18. Delete account

NOTE: For permissions see <https://docs.google.com/spreadsheets/d/1wHPDfGIJjxQoZR0tun432C2QtJbNpFmWuPXgzhOZieI/edit#gid=1965868446>

Request:

{ accountDelete:

{

id: "hash",

}

}

Answer:

{ accountDelete: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 19. Delete factory

NOTE: Only Enterprise user if owner, otherwise only admins

Request:

{ factoryRemove:

{

id: "hash",

}

}

Answer:

{ factoryRemove: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 20. Delete enterprise

NOTE: Only Enterprise user if owner, otherwise only admins

Request:

{ enterpriseRemove:

{

id: "hash",

}

}

Answer:

{ enterpriseRemove: { done: true } }

Errors:

{ errorCode: "notLoggedIn", error: "Can not perform action without logging in" }

## 21. Save map settings

Request:

{"saveMapSettings":

{

"centerX": 1234.5678,

"centerY": 8765.4321,

"zoomLevel": 10.0,

"map": "**OSM map"**,

"layers": ["Provinces", "Communes"]

}

}

Response:

{"done": True}

## 22. Load map settings

Request:

{}

Response:

{"loadMapSettings":

{

"load": true, // NOTE: False if no data to load

"centerX": 1234.5678, // Only if load: True

"centerY": 8765.4321, // Only if load: True

"zoomLevel": 10.0, // Only if load: True

"map": "**OSM map"**, // Only if load: True

"layers": ["Provinces", "Communes"] // Only if load: True

}

}

## 23. Search factories (for reports)

NOTE: As per <https://qa.simosol.fi/issues/4587> you can have only one search limiter (ids, region, province, district, geom, search) except that "search" can be in addition to region, province, district or geom. Likewise you \_must\_ have at least one search limiter.

NOTE: Only for managers and enterprise users

NOTE: Freeform polygon only for managers

NOTE: As per <https://qa.simosol.fi/issues/4929> the search now functions with lists

Request:

{"factorySearch":

{

"search": [{"field": "...", “values”: [{"value": "...", operator: "="}]}], // optional

/\*field is one of ('industryType', 'industryGroup',

'classByOwner', 'certificate', 'ocsClass',

'designedInputCapacity',

'designedOutputCapacity', 'totalStaff',

'productType', 'certifiedWoodInput',

'certifiedWoodOutput', 'importedWoodInput',

'exportedOutput')

Operator is one of ("=", "<", ">")

Value is code for matching fields, integer for designedXCapacity and totalStaff

and "0" or "1" for \*Input and \*Output booleans.

\*/

"geom": {"coordinates": [[[625900.0, 1184800.0],

[625900.0, 1184900.0],

[625930.0, 1184900.0],

[625930.0, 1184800.0],

[625900.0, 1184800.0]]],

"type": "Polygon"}, // optional, ONLY managers -- geojson

"ids": ["factory id", "factory id"], // optional, ONLY managers

"region": "region code", // optional

"province": "region code", // optional

"district": "region code", // optional

“ownedOnly”: “1” // optional, either “0” or “1”, filters to only owned factories

}

}

Response:

{"factorySearch":

{

"count": 1 // amount of results (since enterprise users only get this info)

"items": [ // Only for manager users

{"id": "uid",

"name": "factory name",

"coordinateX": 12345.67,

"coordinateY": 12345.67,

"regionName": "...",

"provinceName": "...",

"districtName": "...",

"communeName": "...",

"industryType": "...",

"industryGroup": "...",

"classByOwner": "...",

"certificate": ["...", ...],

"ocsClass": "...",

"designedInputCapacity": 123.56,

"designedInputCapacityUnit": "...",

"designedOutputCapacity": 123.56,

"designedOutputCapacityUnit": "...",

"productType": "..."}

]

}

}

## 24. Search factories for monitoring report

{"factorySearchMonitoring": {

"geom": {"coordinates": [[[625900.0, 1184800.0],

[625900.0, 1184900.0],

[625930.0, 1184900.0],

[625930.0, 1184800.0],

[625900.0, 1184800.0]]],

"type": "Polygon"}, // optional, ONLY managers, RESTRICTED to owned -- geojson

"ids": ["factory id", "factory id"], // optional, ONLY owned/owned area

“enterpriseIds”: [“enterprise id”, “enterprise id”], //optional, ONLY owned/owned area

"region": "region code", // optional, ONLY managers, ONLY owned

"province": "region code", // optional, ONLY managers, ONLY owned

"district": "region code" // optional, ONLY managers, ONLY owned

}}

Response:

{"factorySearchMonitoring":

{

"count": 1 // amount of results

"items": [

{"id": "uid",

"name": "factory name",

"coordinateX": 12345.67,

"coordinateY": 12345.67,

"regionName": "...",

"provinceName": "...",

"districtName": "...",

"communeName": "...",

"industryType": "...",

"industryGroup": "...",

"classByOwner": "...",

"certificate": ["...", ...],

"ocsClass": "...",

"designedInputCapacity": 123.56,

"designedInputCapacityUnit": "...",

"designedOutputCapacity": 123.56,

"designedOutputCapacityUnit": "...",

"productType": "..."}

]

}

}

## 25. Get quick report in excel file

Request:

{"reportQuickGet": {

same as in factorySearch +

withForestResourceData: “1”, // optional, “1” or “0” boolean to include in report

fromMonth: number,

fromYear: number,

toMonth: number,

toYear: number,

}}

Response:

{"reportQuickGet": {“reportFile”: “<filename\_on\_server>”}}

The report is available for one minute from creation to download from the <filename\_on\_server> after which it is removed (no pre-path, just <ip\_or\_address>/filename).

Fetch the file with GET

## 26. Get monitoring report in excel file

Request:

{"reportMonitoringGet": {

same as in factorySearchMonitoring +

fromMonth: number,

fromYear: number,

toMonth: number,

toYear: number,

}}

Response:

{"reportMonitoringGet": {“reportFile”: “<filename\_on\_server>”}}

The report is available for one minute from creation to download from the <filename\_on\_server> after which it is removed (no pre-path, just <ip\_or\_address>/filename).

Fetch the file with GET

## 27. Set reminder settings

Request:

{"reminderSet":

{

“dayOfMonth”: <1-31>,

“reminderCount”: <0-31>,

“reminderInterval”: <1-31>,

“reminderHour”: <0-23>,

“subject”: “reminder email subject”,

“message”: “reminder email message”

}

}

Response:

{"done": True}

## 28. Get reminder settings

Request:

{}

Response:

{"reminderGet":

{

“dayOfMonth”: <1-31>,

“reminderCount”: <0-31>,

“reminderInterval”: <1-31>,

“reminderHour”: <0-23>,

“subject”: “reminder email subject”,

“message”: “reminder email message”

}

}

## 29. Inspection report upload

Request:

{“inspectionReportUpload”:

{“factory\_id”: <factory uid>,

“month”: 1,

“year”: 2018,

“file”: <base64 encoded pdf file>}

}

Response:

{"inspectionReportUpload":

{“done”: true}

}

## 30. Inspection report download

Request:

{“inspectionReportGet”: {“id”: <report uid>}}

Response:

{“inspectionReportGet”: {“reportFile”: “<filename\_on\_server>”}}

The report is available for one minute from creation to download from the <filename\_on\_server> after which it is removed (no pre-path, just <ip\_or\_address>/filename).

Fetch the file with GET

## 31. Inspection report listing

Request:

{“inspectionReportList”: {“factory\_id”: <factory uid>}}

Response:

{

"inspectionReportList":

{

“reports”: [{“id”: <report uid>, “year”: 2018, “month”: 1}]

}

}

## 32. Fetching admin area bounds

Request:

{“boundsGet”: {

“level”: <one of “commune”, “district”, “province”, “region”>,

“code”: “123”

}}

Response:

{

“boundsGet”: {“bounds”: [<minx>, <miny>, <maxx>, <maxy>]}

}

Errors:

"invalidValue": "Received invalid value {} for key {}",

"geoServerConnError": "Connection to FIMS geoserver failed",

"geoServerError": "Failed to get valid data from FIMS geoserver",