# Express NodeJS alkalmazás

A projekt létrehozása az Express Application Generator eszközzel történt.

express vizkeszlet-gazdalkodas-app --view=pug --css=sass

create : vizkeszlet-gazdalkodas-app

create : vizkeszlet-gazdalkodas-app/package.json

create : vizkeszlet-gazdalkodas-app/app.js

create : vizkeszlet-gazdalkodas-app/routes

create : vizkeszlet-gazdalkodas-app/routes/index.js

create : vizkeszlet-gazdalkodas-app/routes/users.js

create : vizkeszlet-gazdalkodas-app/public

create : vizkeszlet-gazdalkodas-app/views

create : vizkeszlet-gazdalkodas-app/views/index.pug

create : vizkeszlet-gazdalkodas-app/views/layout.pug

create : vizkeszlet-gazdalkodas-app/views/error.pug

create : vizkeszlet-gazdalkodas-app/bin

create : vizkeszlet-gazdalkodas-app/bin/www

create : vizkeszlet-gazdalkodas-app/public/images

create : vizkeszlet-gazdalkodas-app/public/javascripts

create : vizkeszlet-gazdalkodas-app/public/stylesheets

create : vizkeszlet-gazdalkodas-app/public/stylesheets/style.sass

install dependencies:

> cd vizkeszlet-gazdalkodas-app && npm install

run the app:

> SET DEBUG=vizkeszlet-gazdalkodas-app:\* & npm start

Megjelenítő motor: PUG

CSS forrás: SASS alapokon, scss kiterjesztésű fájlokkal

# MSSQL használata

$ npm install --save mssql

Minden adatbázistáblára egy modell osztályt készítünk, amivel az adatbázisból megvalósítjuk a lekérdezéseket, mentéseket, frissítéseket és törléseket.

# Modell eredmény adatok betöltése

A modellfuttatás .dss fájlban tárolt eredményeit a DSSVue alkalmazással megnyitva a következő (Flow, Stage, Location elevation, Location flow, Vízkészlet, Vízkivételek, Vízbeeresztések, Gate opening, Vízkészlet változás hossz-szelvény) típusú értékeket kell Excel-be és onnan .csv kiterjesztésű fájlba menteni. A betöltő program…

A fájlok feltöltésére a formidable node module-t használom.

A time intervalt az exportált csv fájlban meg kell adni mert a DSSVue nem tölti ki.

A meta\_data elemek törlése az elemeikkel cascade-ra van állítva az adatbázisban. Így a meta\_data törlése törli a hozzá kapcsolódó bejegyzéseket:

* FLOW
* FLOW-CUM
* STAGE
* ….

# Sequelize ORM használata

<http://docs.sequelizejs.com/>

<https://groundberry.github.io/development/2016/11/04/build-your-node-app-with-express-and-sequelize.html>

$ npm install --save sequelize

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises>

<https://www.sitepoint.com/node-js-mvc-application/>

<https://vooban.com/en/tips-articles-geek-stuff/how-to-quickly-create-a-simple-rest-api-for-sql-server-database/>

<https://vmokshagroup.com/blog/building-restful-apis-using-node-js-express-js-and-ms-sql-server/>

ESRI Javascript API

<https://geoportal.vizugy.hu/arcgis/rest/services>

<https://dojotoolkit.org/documentation/tutorials/1.10/modern_dojo/>

<https://developers.arcgis.com/javascript/3/jshelp/inside_dojo.html>

<https://developers.arcgis.com/javascript/3/jsapi/>

<https://developers.arcgis.com/documentation/core-concepts/rest-api/>

<https://developers.arcgis.com/javascript/3/jsapi/argument_aliases.html>

EZEKET KIPRÓBÁLTAM DE NEM VOLT TÖKÉLETES

# MSSQL használata normál modellekkel

<http://timjrobinson.com/how-to-structure-your-nodejs-models-2/>