AGENDA

- **Development tools**
- **Install and deploy**
- **Demonizing the application**
- **Logging**

DEVELOPMENT TOOLS. ESLINT

- De-facto standart
- Checking possible code errors
- Fixing code style
- Detect code anti-patterns
- Rules are pluggable
- Supports typescript also
- Integration with IDE

https://eslint.org/

DEVELOPMENT TOOLS. ESLINT. Config

```javascript
module.exports = {
    parser: '@typescript-eslint/parser', // Specifies the ESLint parser
    extends: [
        'plugin:@typescript-eslint/recommended', // Uses the recommended rules from the @typescript-eslint/eslint-plugin
    ],
    parserOptions: {
        ecmaVersion: 2018, // Allows for the parsing of modern ECMAScript features
        sourceType: 'module', // Allows for the use of imports
    },
    rules: {
        // Place to specify ESLint rules. Can be used to overwrite rules specified from the extended configs
        // e.g. "@typescript-eslint/explicit-function-return-type": "off",
    },
};
```

https://eslint.org/

DEVELOPMENT TOOLS. PRETTIER

- Code formatter
- Working with ESLint

DEVELOPMENT TOOLS. PRETTIER

```javascript
module.exports = {
    semi: true,
    trailingComma: 'all',
    singleQuote: true,
    printWidth: 120,
    tabWidth: 4,
};
```

https://prettier.io/

DEVELOPMENT TOOLS. PRETTIER

```
module.exports = {
    parser: '@typescript-eslint/parser', // Specifies the ESLint parser
    extends: [
        'plugin:@typescript-eslint/recommended', // Uses the recommended rules from the @typescript-eslint/eslint-plugin
        'prettier/@typescript-eslint', // Uses eslint-config-prettier to disable ESLint rules from @typescript-eslint/eslint-plugin that
        'plugin:prettier/recommended', // Enables eslint-plugin-prettier and displays prettier errors as ESLint errors. Make sure this
    ],
    parserOptions: {
        ecmaVersion: 2018, // Allows for the parsing of modern ECMAScript features
        sourceType: 'module', // Allows for the use of imports
    },
};
```

https://prettier.io/

DEVELOPMENT TOOLS. NODEMON

- Automatically restarting the node application
- Doesn't require any additional changes to your code

https://nodemon.io/

DEVELOPMENT TOOLS. NODEMON

```json
// nodemon.json
{
  "verbose": true,
  "ignore": ["*.test.js", "fixtures/*"],
....
}
```
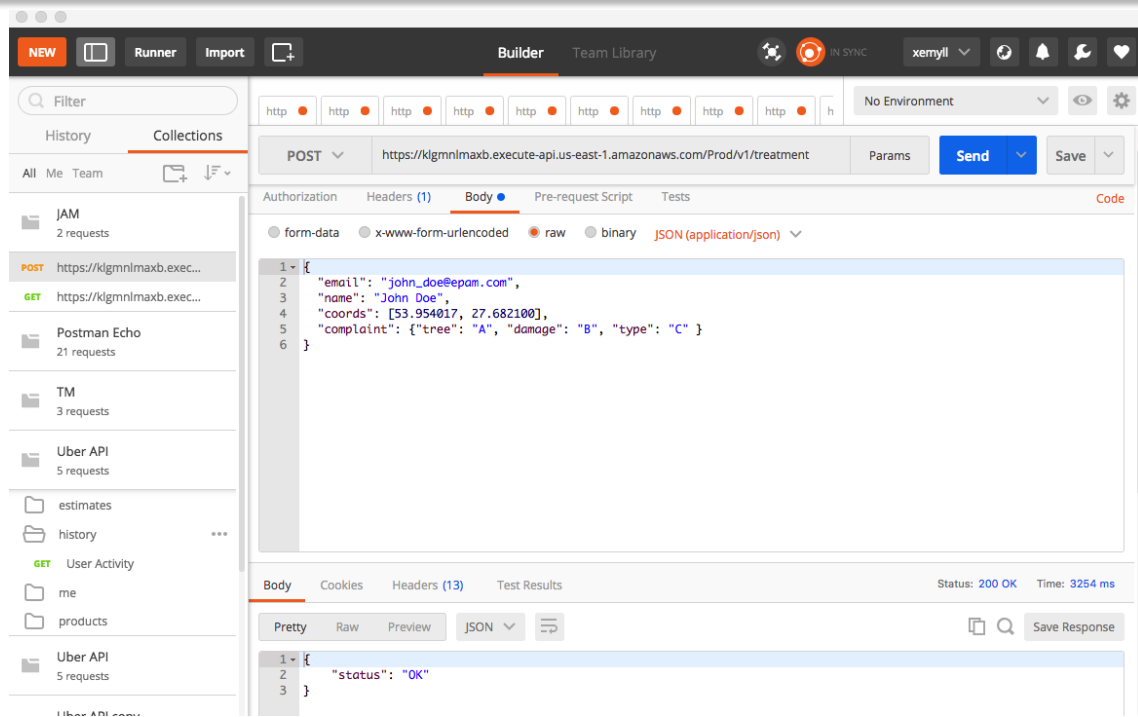
https://nodemon.io/

DEVELOPMENT TOOLS. POSTMAN

- Composing HTTP requests (headers, auth, data)

- Automated testing with collections of requests

- Mock specific server data

- History logging

- API monitoring

https://www.getpostman.com

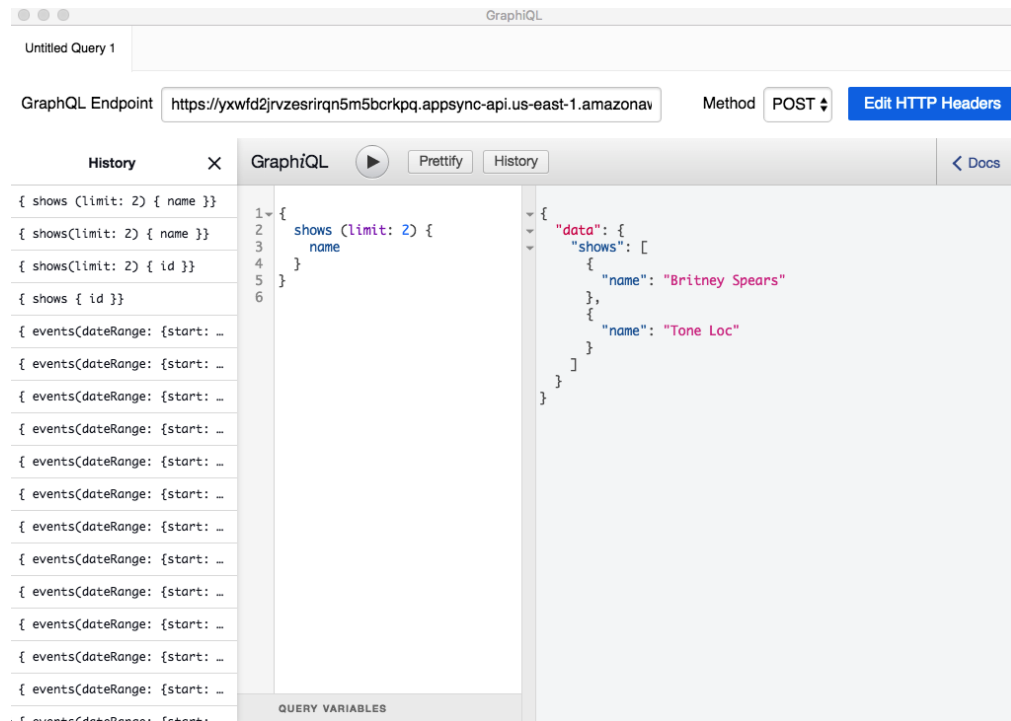# DEVELOPMENT TOOLS. POSTMAN

https://www.getpostman.com

DEVELOPMENT TOOLS. GRAPHIQL

- Syntax highlighting

- Intelligent type ahead

- Real-time error highlighting and reporting

- Automatic query completion

- Run and inspect query results

https://electronjs.org/apps/graphiql

DEVELOPMENT TOOLS. GRAPHIQL

https://electronjs.org/apps/graphiql

DEVELOPMENT TOOLS. FIDDLER

- Request proxying

- Traffic recording (save log with whole requests/responses data)

- Composing HTTP requests (headers, auth, data)

- Performance, security testing

- Customizable free tool

https://www.telerik.com/fiddler

INSTALL AND DEPLOY. NVM

- Node version manager
- CLI

https://github.com/creationix/nvm

NVM

INSTALL AND DEPLOY. DOCKER

- Build Once & Run Anywhere
- Safe runtime environment for app
- No missing dependencies
- Automate testing, integration, packaging

https://www.docker.com/

DEMONIZING THE APP. PM2

- Process manager
- Clustering
- Monitoring
- Logging

https://github.com/Unitech/pm2

DEMONIZING THE APP. FOREVER

Minimalistic and simple

https://github.com/foreverjs/forever

LOGGING. WINSTON

- Streams
- Transport
- Formatting
- Levels

https://github.com/winstonjs/winston

LOGGING. BUNYAN

JSON logging library

https://github.com/trentm/node-bunyan

NODE.JS GLOBAL

**DEPLOY AND TOOLS
BY
SIARHEI MELNIK**