Modules

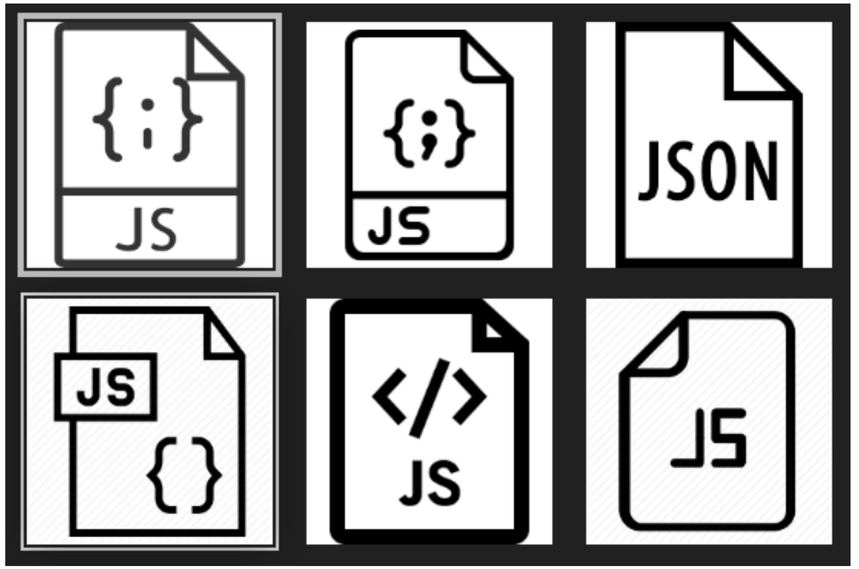
Modules

PLAY VIDEO

The backend will use a modular approach, relying on specific mechanism to import/export shared objects

Javascript Modules

- To structure an application coherently, the backend consists of separate Javascript files.
- Objects declared in these files must be
  - exported by one file
  - imported by another
- In order to keep each module focused on a specific responsibility



Example

- 2 separate object defined in a single file
- Methods called on these objects at the end of the file

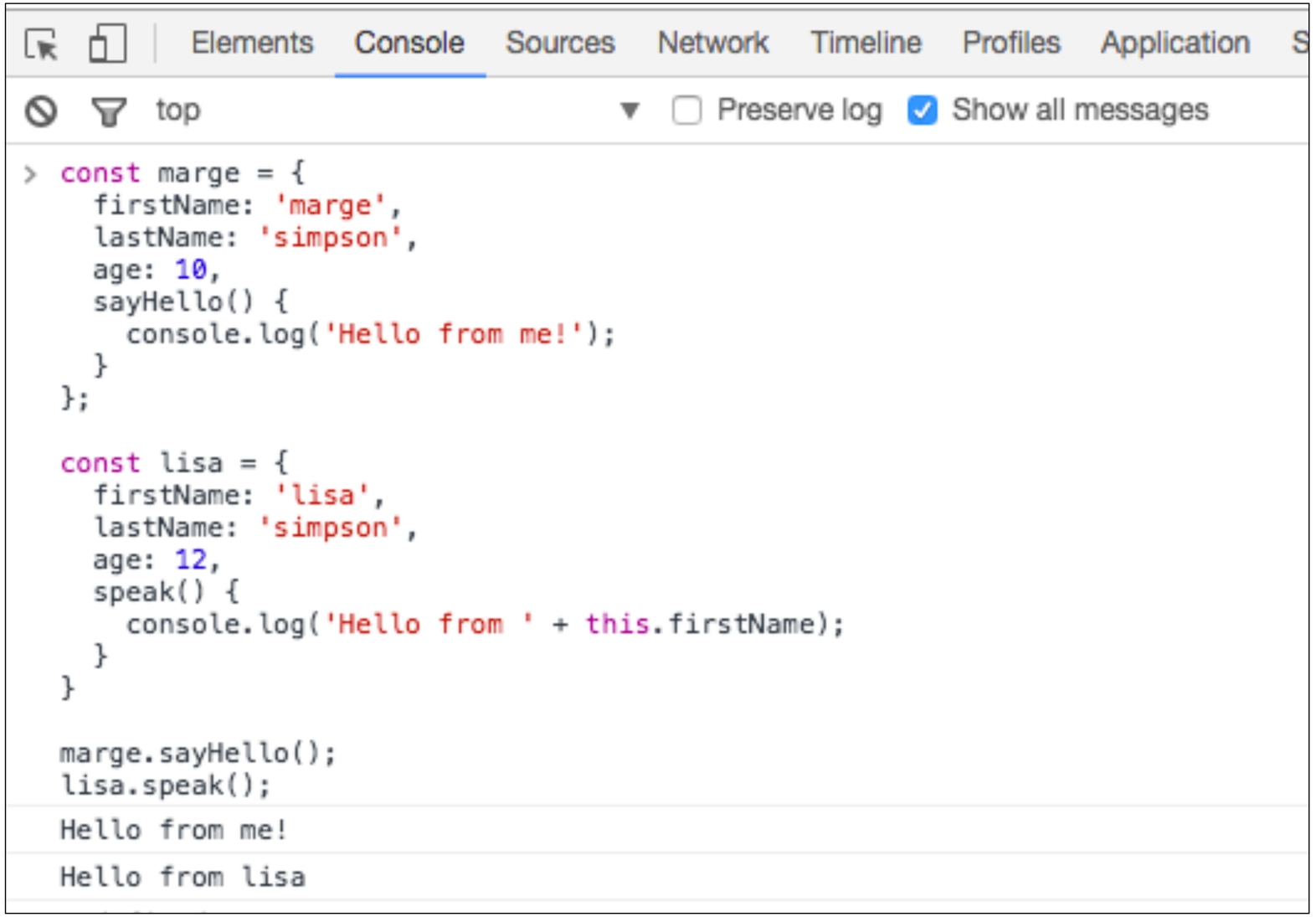
standalone.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ' + this.firstName);
  }
}

marge.sayHello();
lisa.speak();
```

In Chrome JS Console



Modularise the Program

standalone.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ‘
                + this.firstName);
  }
}

marge.sayHello();
lisa.speak();
```

marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};
```

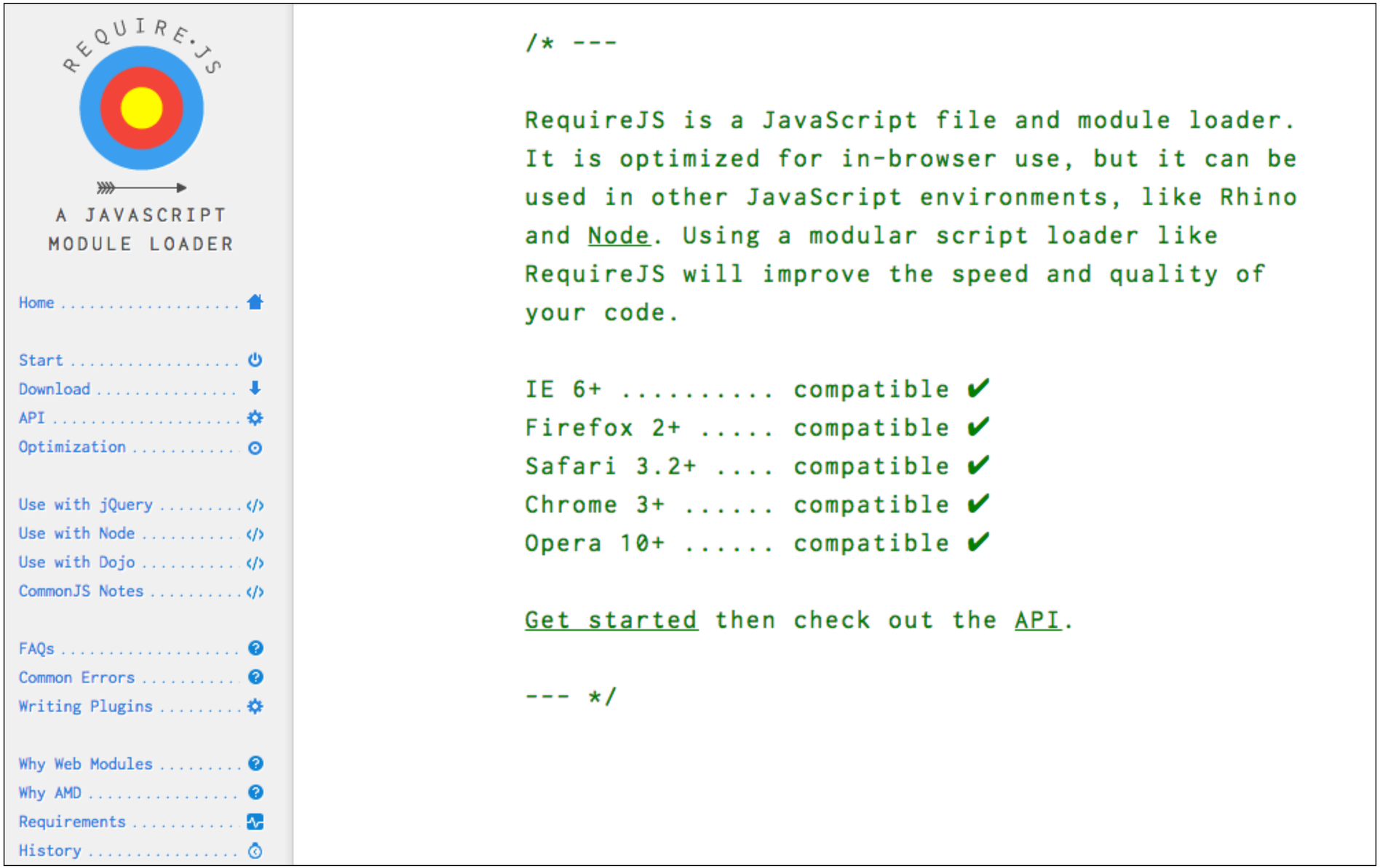
lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ‘
                + this.firstName);
  }
}
```

main.js

```
marge.sayHello();
lisa.speak();
```

http://requirejs.org/



Modularise the Program

- These three modules:
  - marge.js
  - lisa.js
  - main.js
- Are completely separate.
- main.js cannot use marge or lisa objects

marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};
```

lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from ‘
                + this.firstName);
  }
}
```

main.js

```
marge.sayHello();
lisa.speak();
```

module.exports

module.exports makes the listed object available to other modules

marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

module.exports = marge;
```

lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from '
                  + this.firstName);
  }
}

module.exports = lisa;
```

require

marge.js

```
const marge = {
  firstName: 'marge',
  lastName: 'simpson',
  age: 10,
  sayHello() {
    console.log('Hello from me!');
  }
};

module.exports = marge;
```

lisa.js

```
const lisa = {
  firstName: 'lisa',
  lastName: 'simpson',
  age: 12,
  speak() {
    console.log('Hello from '
                  + this.firstName);
  }
}

module.exports = lisa;
```

require identifies and imports objects defined in other modules

main.js

```
const marge = require('./marge.js');
const lisa = require('./lisa.js');

marge.sayHello();
lisa.speak();
```

Modules in back-end - Example

```
controllers/about.js
controllers/dashboard.js
utils/logger.js
routes.js
server.js
```

- Each of these modules will use export and require to establish dependencies

- 5 separate modules
  - sever.js
  - routes.js
  - about.js
  - dashboard.js
  - logger.js