

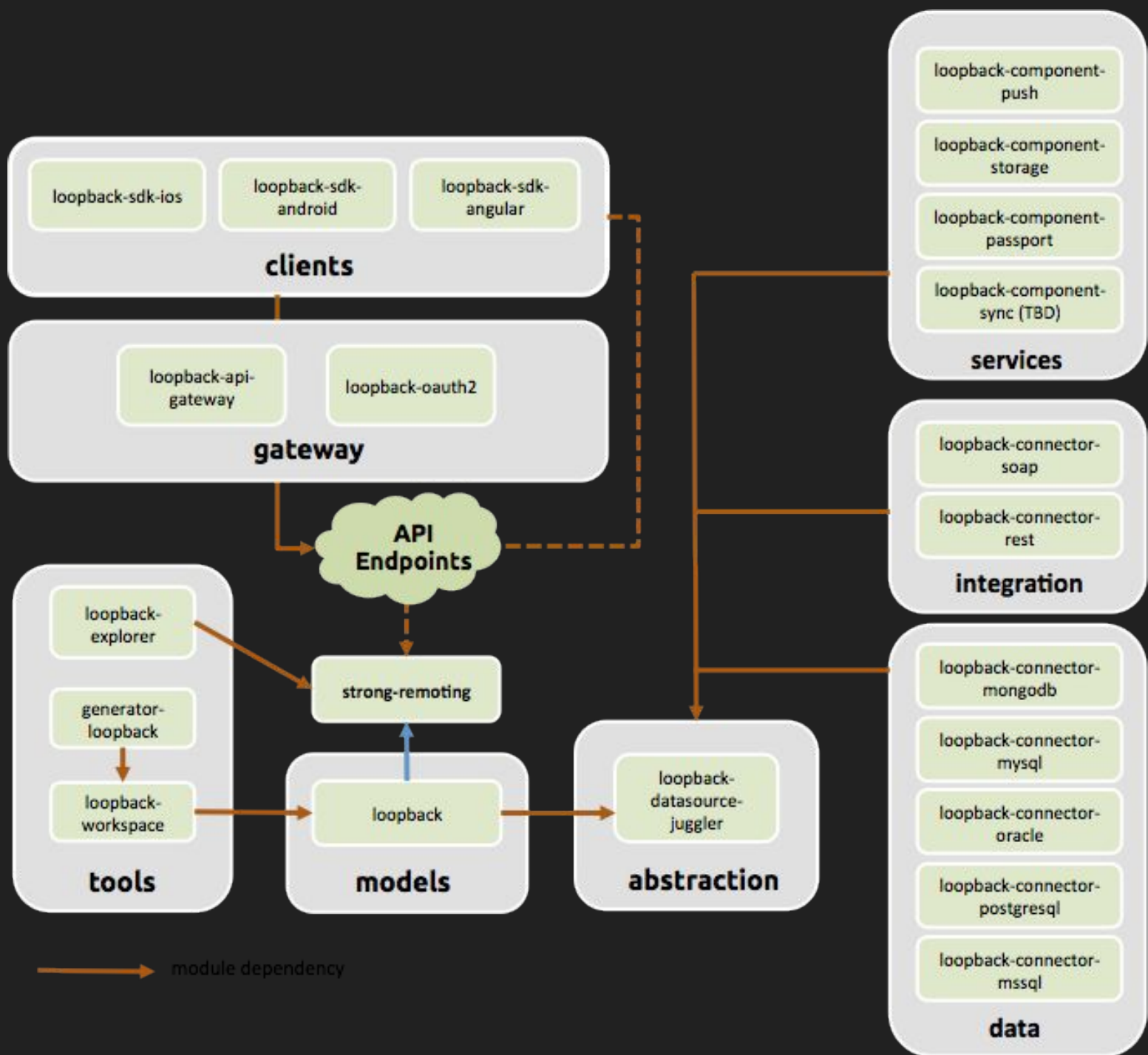


LoopBack

Vincent Wagner

LoopBack

- <https://loopback.io/>
 - Node.js-Rest-API-Framework
 - Basiert auf Express.js
 - Supportet von StrongLoop (IBM)
-
- ORM (Postgresql, Mondodb, ...)
 - Benutzer- und Gruppenverwaltung und ACL
 - Verschiedene Client-APIs (AngularJS, Android, iOS, ...)
 - Mächtige Codegeneratoren (yeoman)



Getting started

```
$ npm install -g strongloop
```

```
#app Generator (benötigt eine Internetverbindung)
```

```
$ slc loopback
```

```
#Starten
```

```
$ node . #Oder slc run -cluster 4
```

```
#Explorer öffnen
```

```
$ open http://localhost:3000/explorer/
```

```
#Generatoren
```

```
$ slc loopback --help
```

```
#Datasource Generator (benötigt eine Internetverbindung)
```

```
$slc loopback:datasource
```

Model Generator

\$ slc loopback:model

Lake

- name: String
- position: Geopos

Migrieren

```
var loopback = require('loopback');  
var boot = require('loopback-boot');  
var app = loopback();
```

```
boot(app, __dirname + '/../server', function(err) {  
  if (err) throw err;  
  var ds = app.dataSources.mysql;  
  ds.automigrate(function () {  
    console.log('migrated');  
    process.exit(0);  
  });  
});
```

Relation Generator

\$ slc loopback:city

City

- name: String

\$ slc loopback:relation

- Lake belongs to a city (operator)

\$ slc loopback:relation

- City has many lakes (lakes mapped by operatorId)

Filter (Where)

[http://localhost:3000/api/lakes?filter\[where\]\[name\]=Baggersee](http://localhost:3000/api/lakes?filter[where][name]=Baggersee)

[http://localhost:3000/api/lakes?filter={\"where\":{\"name\":\"Baggersee\"}}](http://localhost:3000/api/lakes?filter={\)

[http://localhost:3000/api/lakes?filter\[where\]\[position\]\[near\]=42,8&filter\[limit\]=1](http://localhost:3000/api/lakes?filter[where][position][near]=42,8&filter[limit]=1)

- Logische Operatoren (AND, OR)
- regexp
- between
- near (Geopoints)
- ...

Weitere Filter

Fields:

[http://localhost:3000/api/lakes?filter\[fields\]\[name\]=true](http://localhost:3000/api/lakes?filter[fields][name]=true)

Include:

[http://localhost:3000/api/cities?filter\[include\]=lakes](http://localhost:3000/api/cities?filter[include]=lakes)

[http://localhost:3000/api/cities?filter\[include\]\[lakes\]\[operator\]=lakes](http://localhost:3000/api/cities?filter[include][lakes][operator]=lakes)

Order:

[http://localhost:3000/api/lakes?filter\[order\]=name](http://localhost:3000/api/lakes?filter[order]=name)

[http://localhost:3000/api/lakes?filter\[order\]=name](http://localhost:3000/api/lakes?filter[order]=name) DESC

Userverwaltung

Schreibrechte verbieten auf allen Modellen

\$ slc loopback:acl

- User.find (nobody)
- User.create (everyone)
- User.login (everyone)
- User.update (owner)

Remote Methoden deaktivieren

```
City.disableRemoteMethod('createChangeStream', true);  
City.disableRemoteMethod('getChangeStream', true);  
City.disableRemoteMethod('count', true);  
City.disableRemoteMethod('__destroyById__lakes', false);
```

AngularJS-Service

\$ lb-ng server.js lb-services.js

- Implementiert \$resource-Methoden

```
lakesControllers.controller(..., ['$scope', 'Lake', function($scope, Lake) {  
    Lake.find({filter:...}).$promise  
        .then(function(){  
            ...});  
}]);
```

<https://loopback.io/>

Vincent.Wagner@posteo.de
<https://github.com/serofax>