

loopback

как перестать “готовить”

НИКОЛАЙ СЛАДКИЙ
MULTIKEY.STUDIO

ДИСКЛЕЙМЕР

Все совпадения с реальными людьми - случайные. Это всё моя больная фантазия и немного юмора (kek). Я не считаю себя каким-то гуру, никого ничему не учу, не хочу никого обидеть и никого ни к чему не призываю. Я просто делясь своими мыслями в своей странной манере. Если что не так - сорян.

НИМАНИЕ! АЖНО!

НИМАНИЕ! АЖНО!

“ПЕРЕСТАТЬ ГОТОВИТЬ” != “ПЕРЕСТАТЬ ДУМАТЬ”

Кто я такой?

Кто я такой?



Кто я такой?



Николай Сладкий

27 лет

2 года в уголовном розыске

4 года в веб-разработке

Где я работаю?

Chief Technical Officer

Где я работаю?

MULTIKEY

Ваше конкурентное преимущество на цифровом рынке

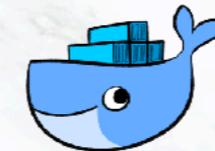
Где я работаю?

MULTIKEY

Ваше конкурентное преимущество на цифровом рынке

Chief Technical Officer

Где я работаю?

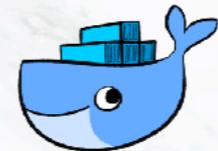


MULTIKEY

Ваше конкурентное преимущество на цифровом рынке

Chief Technical Officer

Где я работаю?



MULTIKEY

Ваше конкурентное преимущество на цифровом рынке

Chief Technical Officer

Где я работаю?



MULTIKEY

Ваше конкурентное преимущество на цифровом рынке



Олег Бартунов

Chief Technical Officer

Чем ещё я занимаюсь?

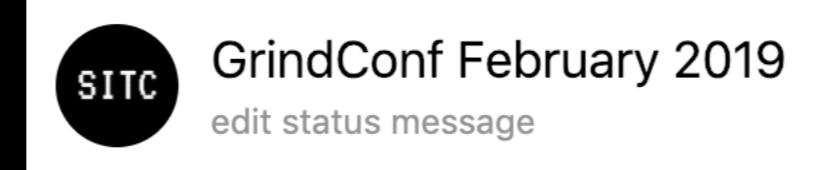
Чем ещё я занимаюсь?

SITE

Чем ещё я занимаюсь?



Samara IT Community



vk.com/samara_it_community

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ТО, ЧТО NODEJS ХОРОШО/ПЛОХО

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ТО, ЧТО NODEJS ХОРОШО/ПЛОХО

ПРО ТО, КАК СДЕЛАТЬ ЧТО-ТО СЛОЖНОЕ

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ТО, ЧТО NODEJS ХОРОШО/ПЛОХО

ПРО ТО, КАК СДЕЛАТЬ ЧТО-ТО СЛОЖНОЕ

ПРО ТО, КАК СДЕЛАТЬ, ЧТОБЫ ЭТО СЛОЖНОЕ НОРМАЛЬНО РАБОТАЛО

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ТО, ЧТО NODEJS ХОРОШО/ПЛОХО

ПРО ТО, КАК СДЕЛАТЬ ЧТО-ТО СЛОЖНОЕ

ПРО ТО, КАК СДЕЛАТЬ, ЧТОБЫ ЭТО СЛОЖНОЕ НОРМАЛЬНО РАБОТАЛО

ПРО “ЛУЧШИЕ ПРАКТИКИ”

ПРО ЧТО ГОВОРИТЬ НЕ БУДЕМ?

ПРО ТО, ЧТО NODEJS ХОРОШО/ПЛОХО

ПРО ТО, КАК СДЕЛАТЬ ЧТО-ТО СЛОЖНОЕ

ПРО ТО, КАК СДЕЛАТЬ, ЧТОБЫ ЭТО СЛОЖНОЕ НОРМАЛЬНО РАБОТАЛО

ПРО “ЛУЧШИЕ ПРАКТИКИ”

И ПРО ТО, ЧТО ТЫ, МОЙ ДОРОГОЙ %USERNAME, КОНЕЧНО ЖЕ СДЕЛАЛ БЫ НАМНОГО ЛУЧШЕ

ПРО ЧТО ГОВОРИТЬ БУДЕМ?

ПРО ЧТО ГОВОРИТЬ БУДЕМ?

ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ПЕРЕСТАТЬ “ГОТОВИТЬ” NODEJS

ПРО ЧТО ГОВОРИТЬ БУДЕМ?

ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ПЕРЕСТАТЬ “ГОТОВИТЬ” NODEJS

ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ПЕРЕСТАТЬ ПИСАТЬ ОДНО И ТО ЖЕ

ПРО ЧТО ГОВОРИТЬ БУДЕМ?

ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ПЕРЕСТАТЬ “ГОТОВИТЬ” NODEJS

ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ПЕРЕСТАТЬ ПИСАТЬ ОДНО И ТО ЖЕ

И ПРО ТО, ЧТО В 2019 ГОДУ ХОЧЕТСЯ ЗАНИМАТЬСЯ РАЗРАБОТКОЙ

почему nodejs?

nodejs - это быстро

nodejs - это быстро

nodejs - это просто

nodejs - это быстро

nodejs - это просто

nodejs - это javascript

nodejs - это быстро

nodejs - это просто

nodejs - это javascript

nodejs - это чтобы быстро и просто на js сделать... ар!

Что мы хотим видеть в арі?

Что мы хотим видеть в арі?

REST

Что мы хотим видеть в api?

REST

DATASOURCE CONNECTORS

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

Что мы хотим видеть в api?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

ENTITY

Что мы хотим видеть в api?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

Что мы хотим видеть в API?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESTOKEN

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESTOKEN

ROLE

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESTOKEN

ROLEMAPPING

ROLE

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

PUSH

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

PUSH

STORAGE

Что мы хотим видеть в api?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

PUSH

STORAGE

OAUTH

Что мы хотим видеть в арі?

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

PUSH

STORAGE

OAUTH

SDK

И так каждый раз!

REST

DATASOURCE CONNECTORS

ORM

DOCUMENTATION

RELATIONS

ENTITY

USER

ACCESSCONTROLLEVEL

ACCESTOKEN

ROLE

ROLEMAPPING

CRUD

SWAGGER

PUSH

STORAGE

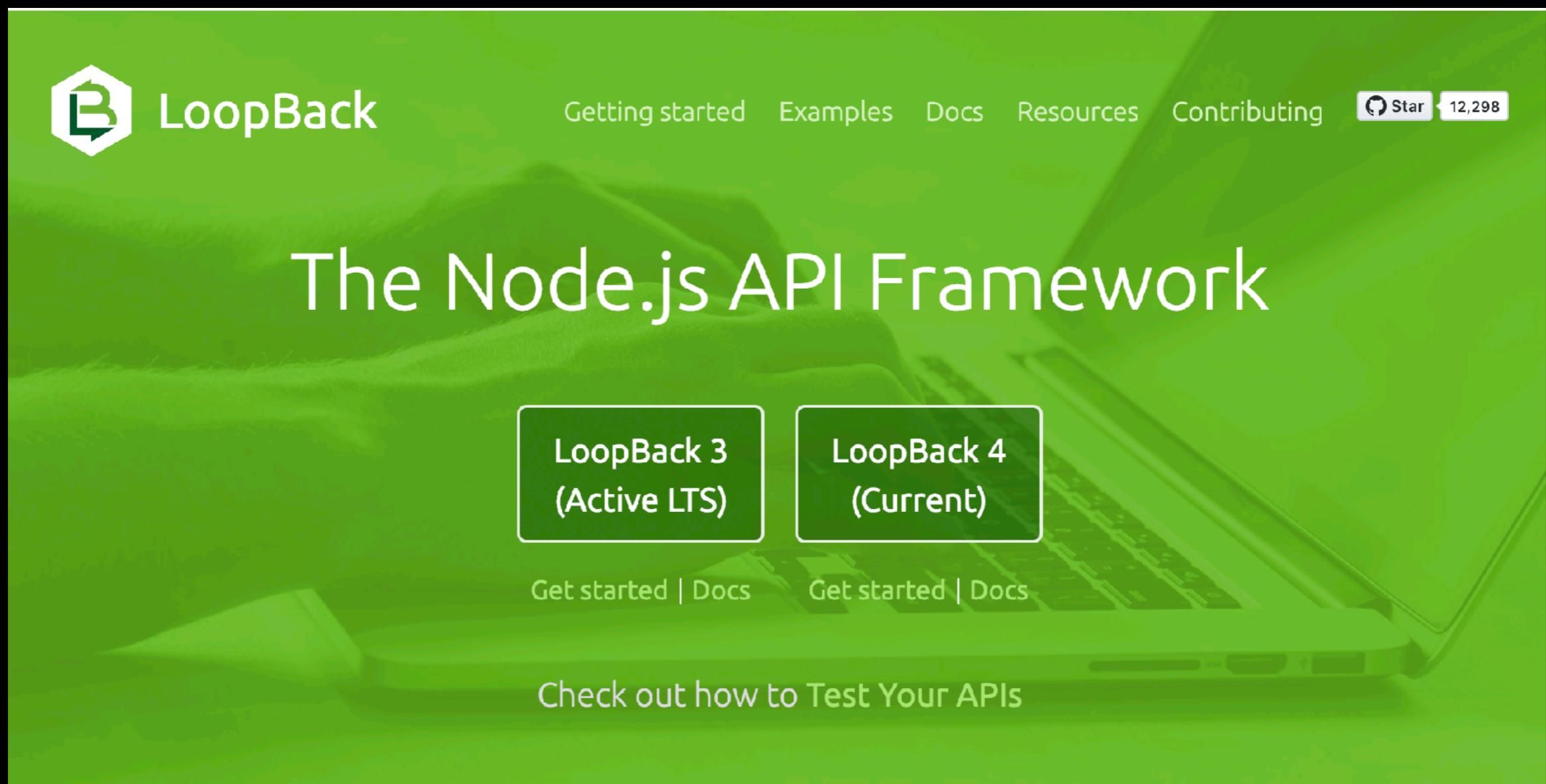
OAUTH

SDK

Вот бы такую штуку, где это всё уже есть!

Вот бы такую штуку, где это всё уже есть!

LOOPBACK



The screenshot shows the LoopBack homepage. At the top left is the LoopBack logo, which consists of a white hexagon with a green 'B' inside. To its right is the word 'LoopBack'. On the far right of the header are navigation links: 'Getting started', 'Examples', 'Docs', 'Resources', and 'Contributing'. Next to 'Contributing' is a GitHub star icon with the number '12,298' next to it. Below the header, the main title 'The Node.js API Framework' is displayed in large white font. Underneath the title are two buttons: 'LoopBack 3 (Active LTS)' and 'LoopBack 4 (Current)'. Below these buttons are two links: 'Get started | Docs' and 'Get started | Docs'. At the bottom center is a call-to-action: 'Check out how to Test Your APIs'.

LoopBack

Getting started Examples Docs Resources Contributing

Star 12,298

The Node.js API Framework

LoopBack 3 (Active LTS)

LoopBack 4 (Current)

Get started | Docs Get started | Docs

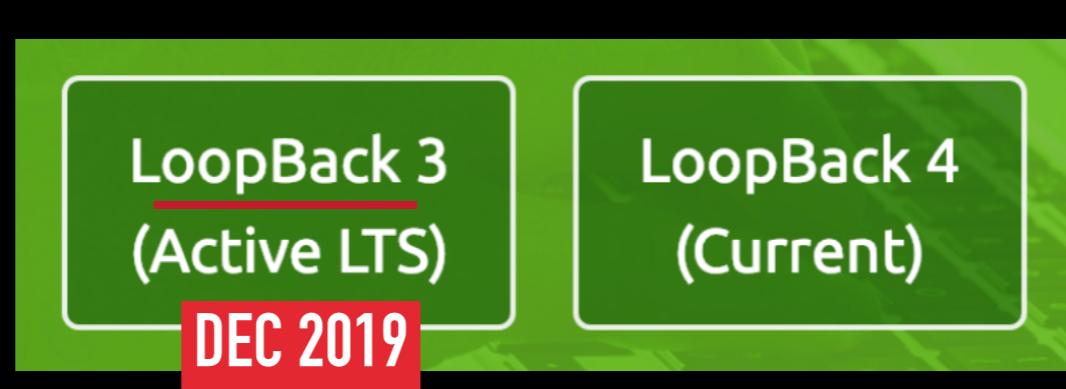
Check out how to Test Your APIs

LoopBack 3
(Active LTS)

LoopBack 4
(Current)

LoopBack 3
(Active LTS)

LoopBack 4
(Current)



*LoopBack 4, the next step in the evolution of LoopBack!
A completely redesigned modular framework especially for API developers.*

About

- ✓ Quickly create dynamic end-to-end REST APIs.
- ✓ Connect devices and browsers to data and services.
- ✓ Use Android, iOS, and AngularJS SDKs to easily create client apps.
- ✓ Add-on components for file management, 3rd-party login, and OAuth2.
- ✓ Runs on-premises or in the cloud

Compare

	LoopBack	Express	Hapi	Sails	Restify	Meteor
Type	API framework	HTTP server library	HTTP server framework	Web MVC framework	REST HTTP library	Full-stack JavaScript app platform
Top Features	Enterprise connectivity, API Explorer, generators, client SDKs, websocket microservices	HTTP routing, middleware	Modularity, security	Rails familiarity, MVC	Simplicity, REST routing	Universal JavaScript, reactive rendering, websocket microservices
Suitable For	Web apps, APIs	Simple web apps	Web apps, APIs	Web apps, APIs	Simple REST APIs	Web apps
Github Stars	8k	29k	7k	16k	6k	36k
Support	IBM/StrongLoop	IBM/StrongLoop	N/A	N/A	N/A	Meteor Development Group
Pure Node runtime	Yes	Yes	Yes	Yes	Yes	No
Client SDKs	Angular, Browser, Node.js, iOS, Android, Xamarin	N/A	None	None	None	JavaScript, Cordova for iOS and Android, React, AngularJS
Export API Definition	Yes	With strong-remoting	None	None	None	With meteor-rest
Tools	API Explorer, CLI code generators	CLI app generator	Yeoman generator	Yeoman generator	Yeoman generator	CLI tool
Extensions	Push, File Storage, Passport, OAuth 2.0, Express Middleware	Express / Connect Middleware	Hapi Plugins			Proprietary package system and repository, npm

Core

Set up models and create REST APIs in minutes

- Easy-to-use CLI wizard
- Create models based on your schema if you have one
- Create dynamic models if you don't
- Built-in API Explorer

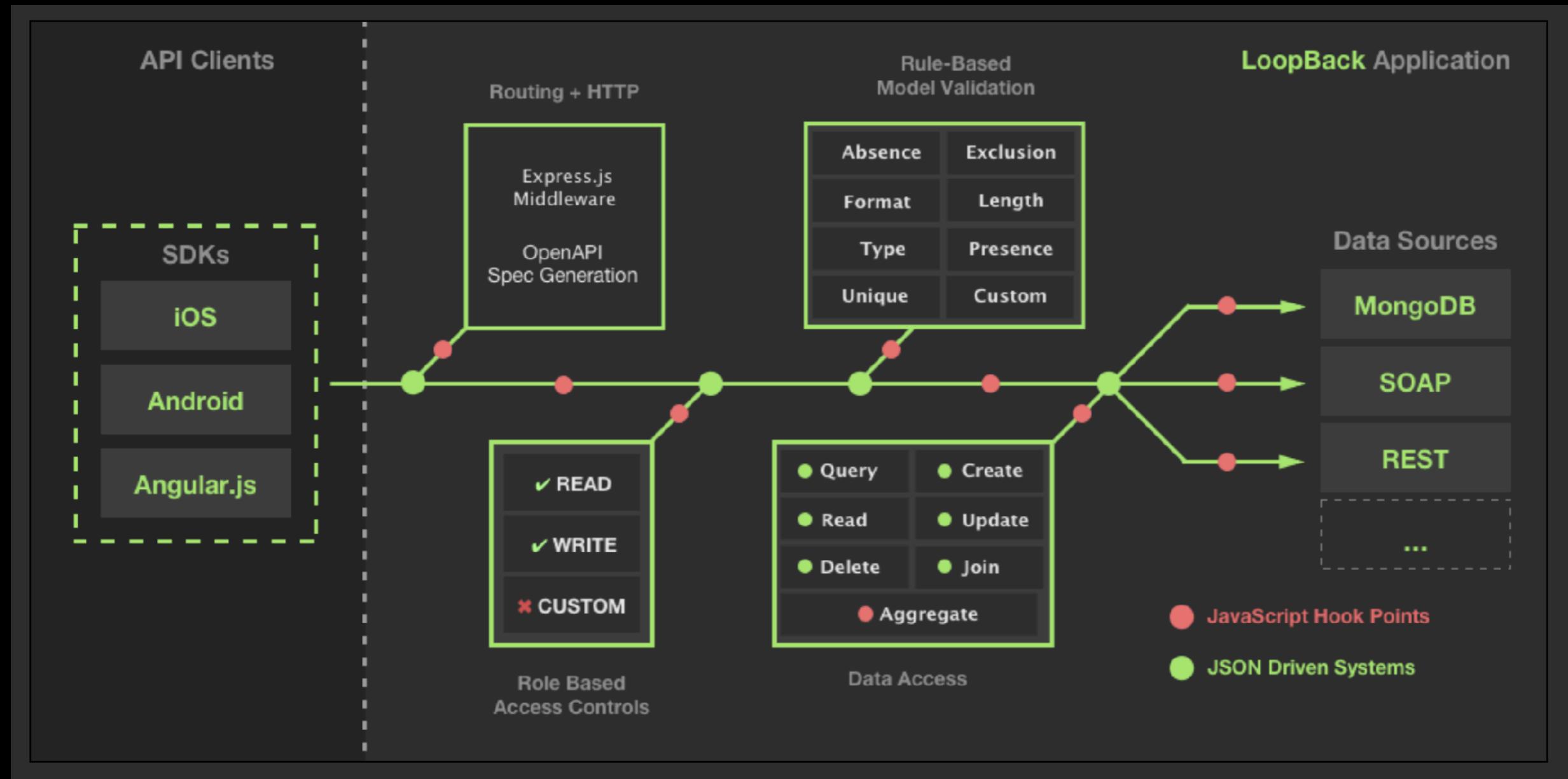
Model relation support

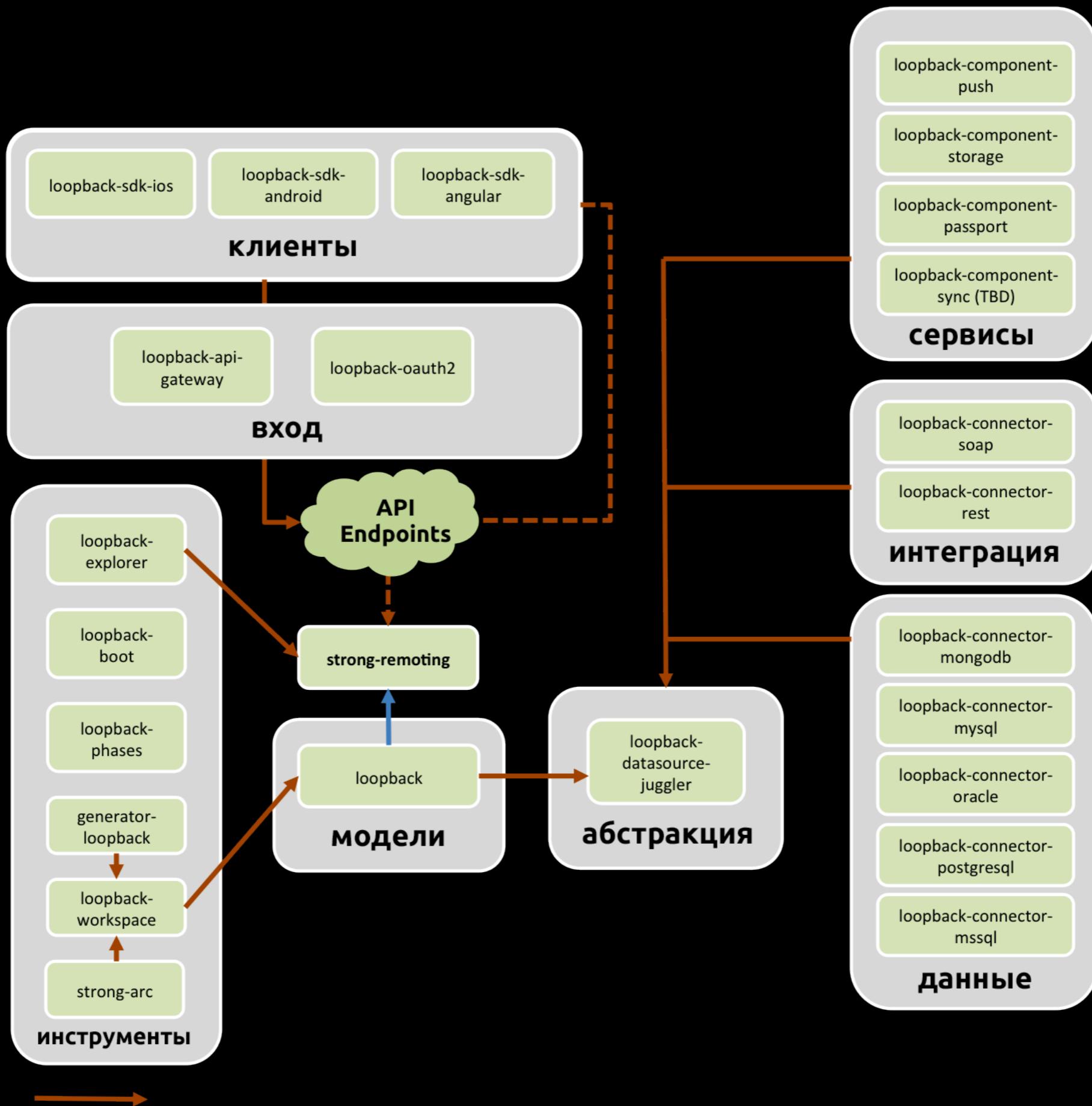
- Define hasMany, belongsTo, hasAndBelongsToMany relationships
- Automatically generates corresponding relational REST endpoints

Easy authentication and authorization setup

- Built-in role-based access controls
- oAuth user and registration models baked in
- Add custom policies using CLI or JSON
- Browser support (JSONP and CORS)

MIT Open-Source license





Juggler

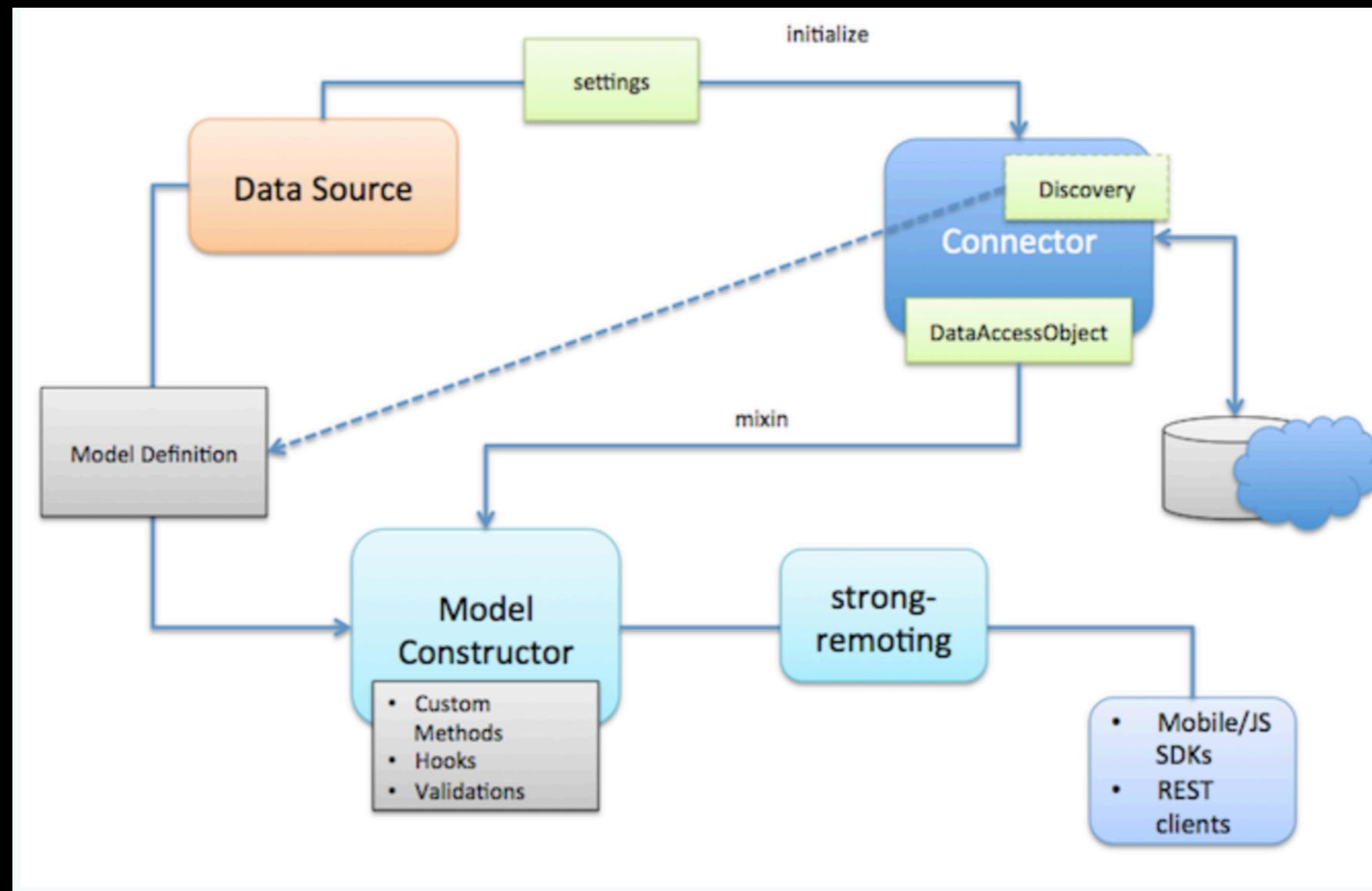
Connect to backend data stores

- MySQL, Oracle, MongoDB, Postgres and others
- Other REST services
- Discovery APIs for Oracle, MySQL, PostgreSQL and SQL Server

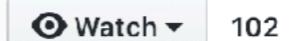
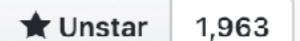
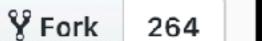
Run Ad-hoc queries

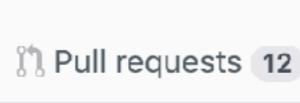
- Run dynamic queries on devices, browsers and the server
- Support for NoSQL-style queries against relational databases

LoopBack DataSource Juggler



 [1602 / jugglingdb](#)

 Watch ▾ 102  Unstar 1,963  Fork 264

 [Code](#)  [Issues 64](#)  [Pull requests 12](#)  [Projects 0](#)  [Wiki](#)  [Insights](#)

Multi-database ORM for nodejs: redis, mongodb, mysql, sqlite3, postgresql, arango, in-memory... <http://1602.github.io/jugglingdb/>

 1,022 commits  9 branches  61 releases  69 contributors



SDK

COMPONENTS

SDK

TRANSACTIONS

COMPONENTS

SDK

TRANSACTIONS

COMPONENTS

QUERYING DATA

SDK

TRANSACTIONS

COMPONENTS

SDK

QUERYING DATA

REMOTE METHODS

TRANSACTIONS

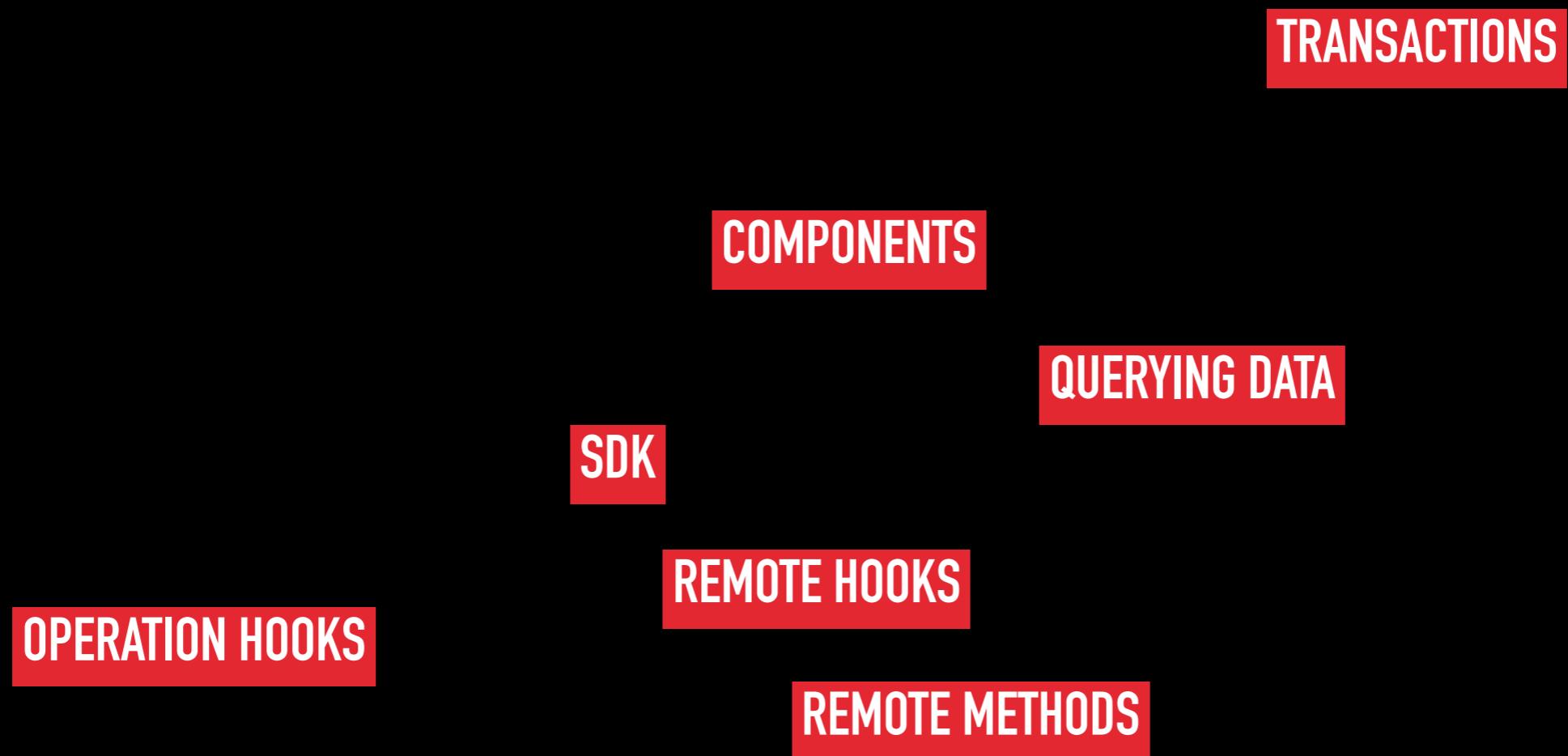
COMPONENTS

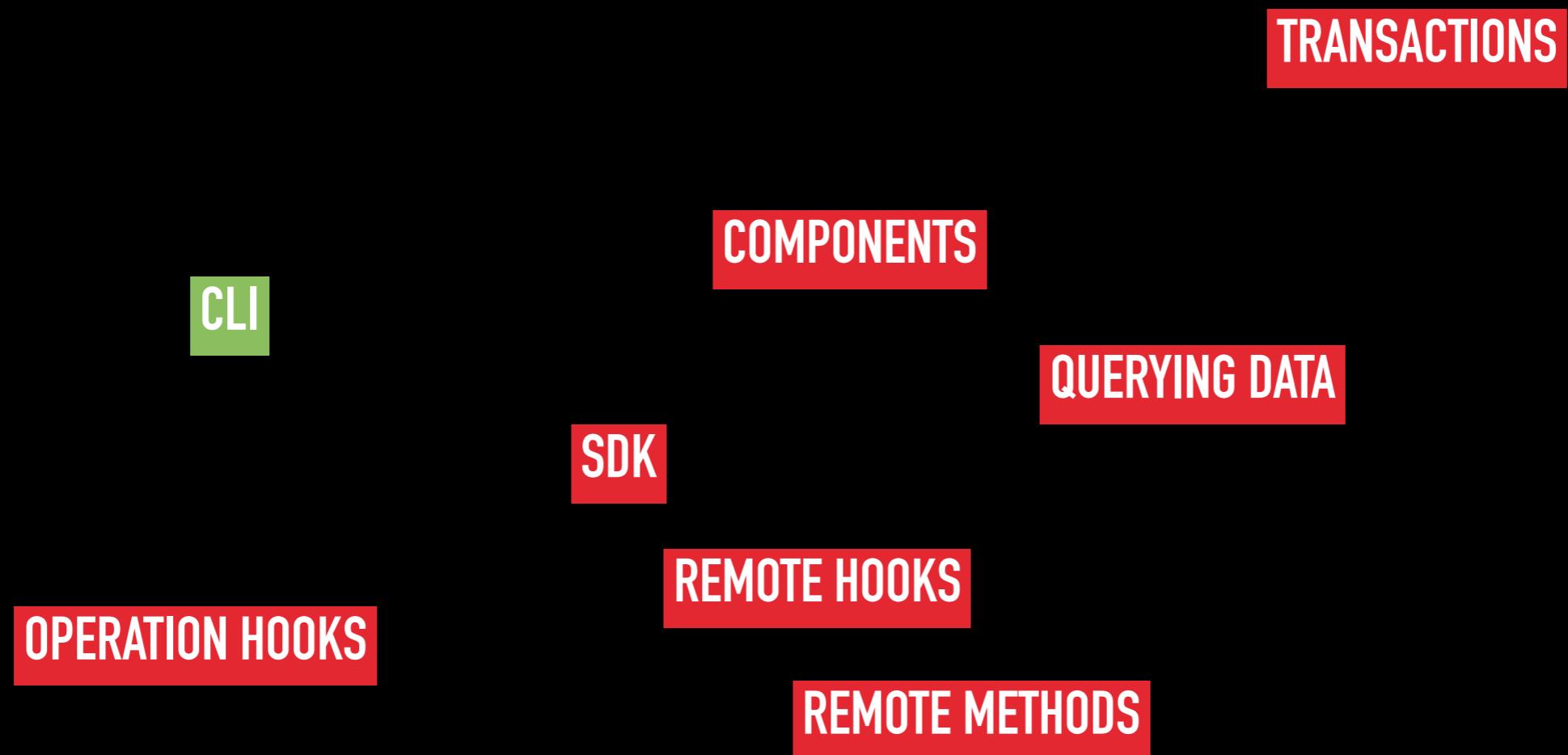
SDK

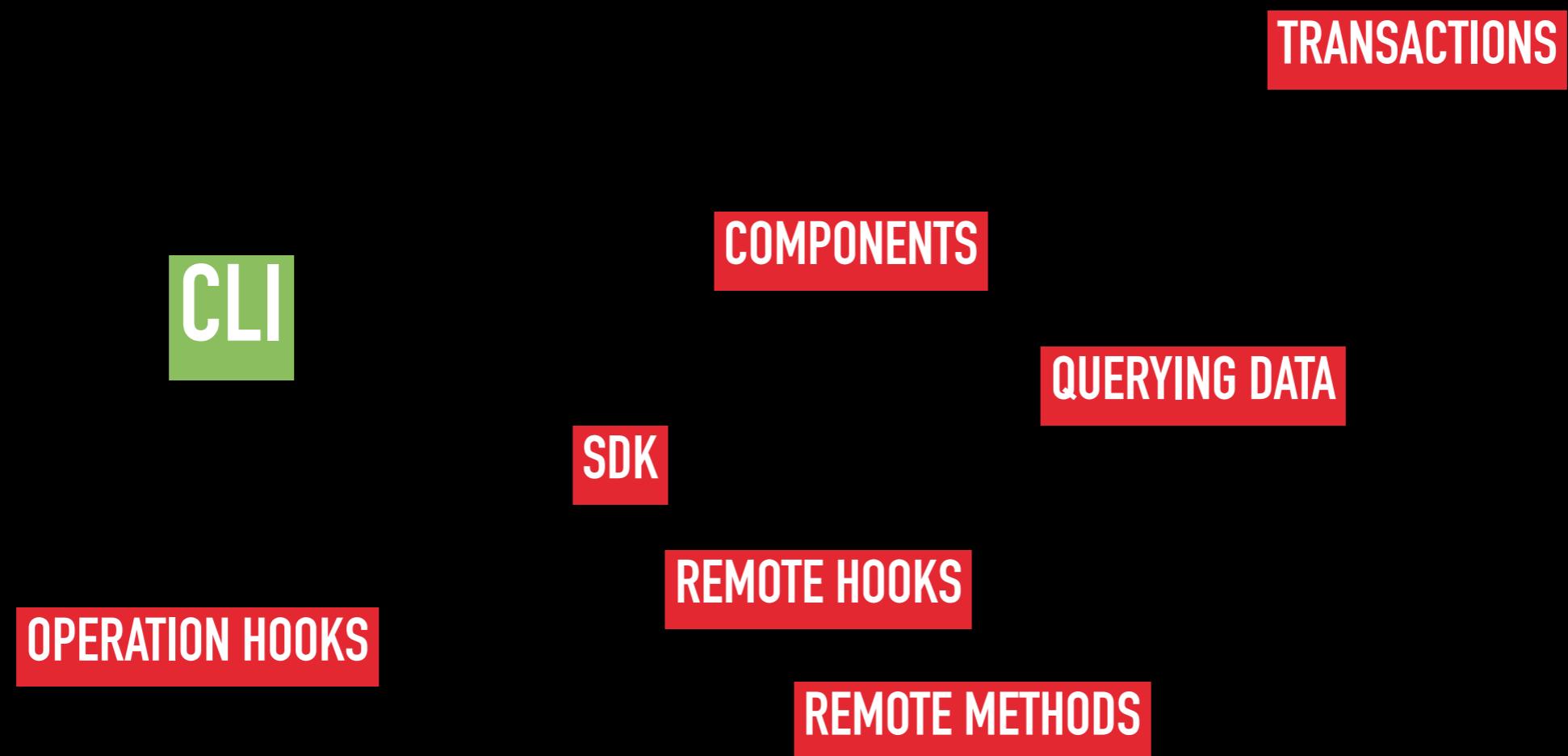
QUERYING DATA

REMOTE HOOKS

REMOTE METHODS

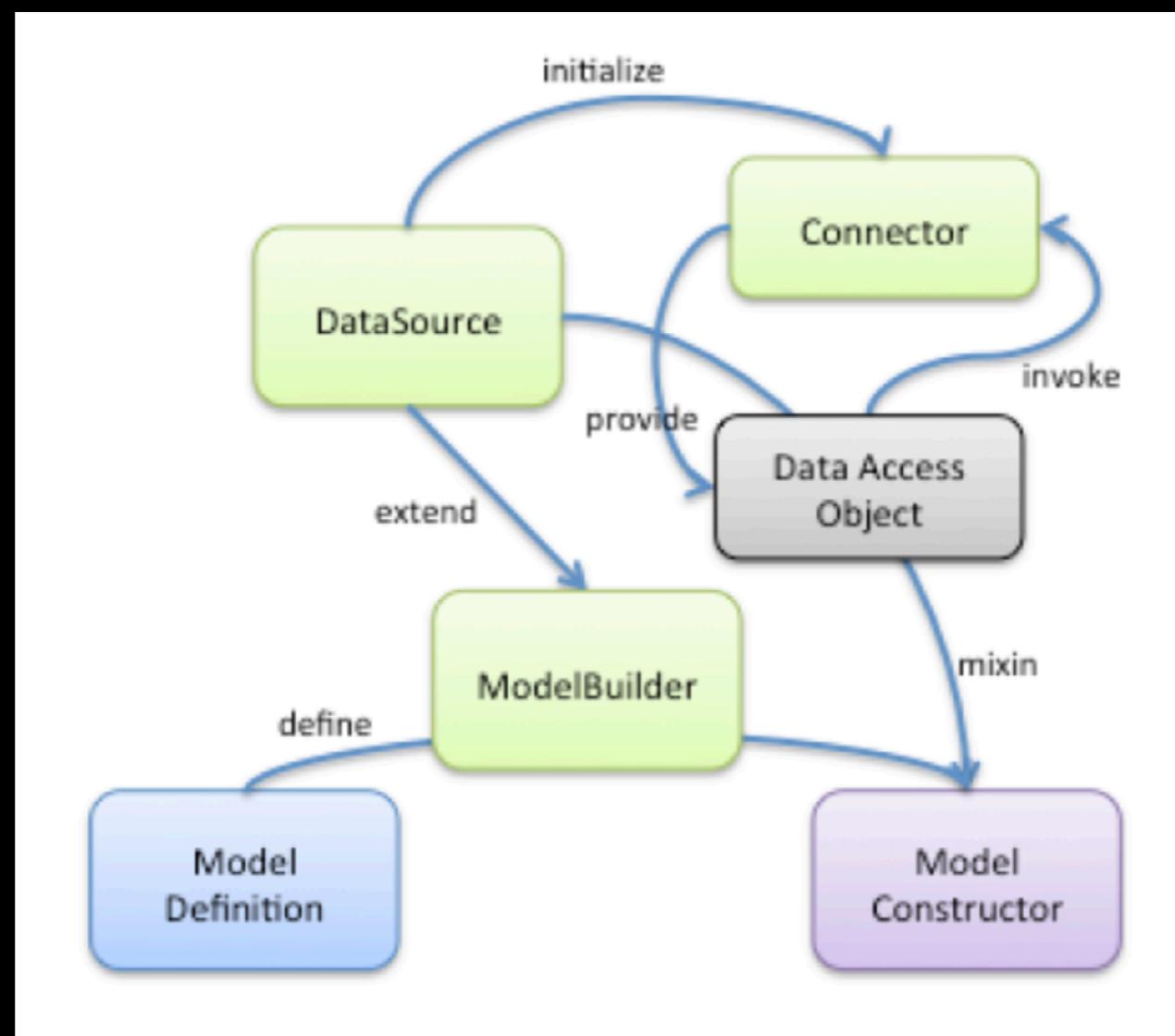






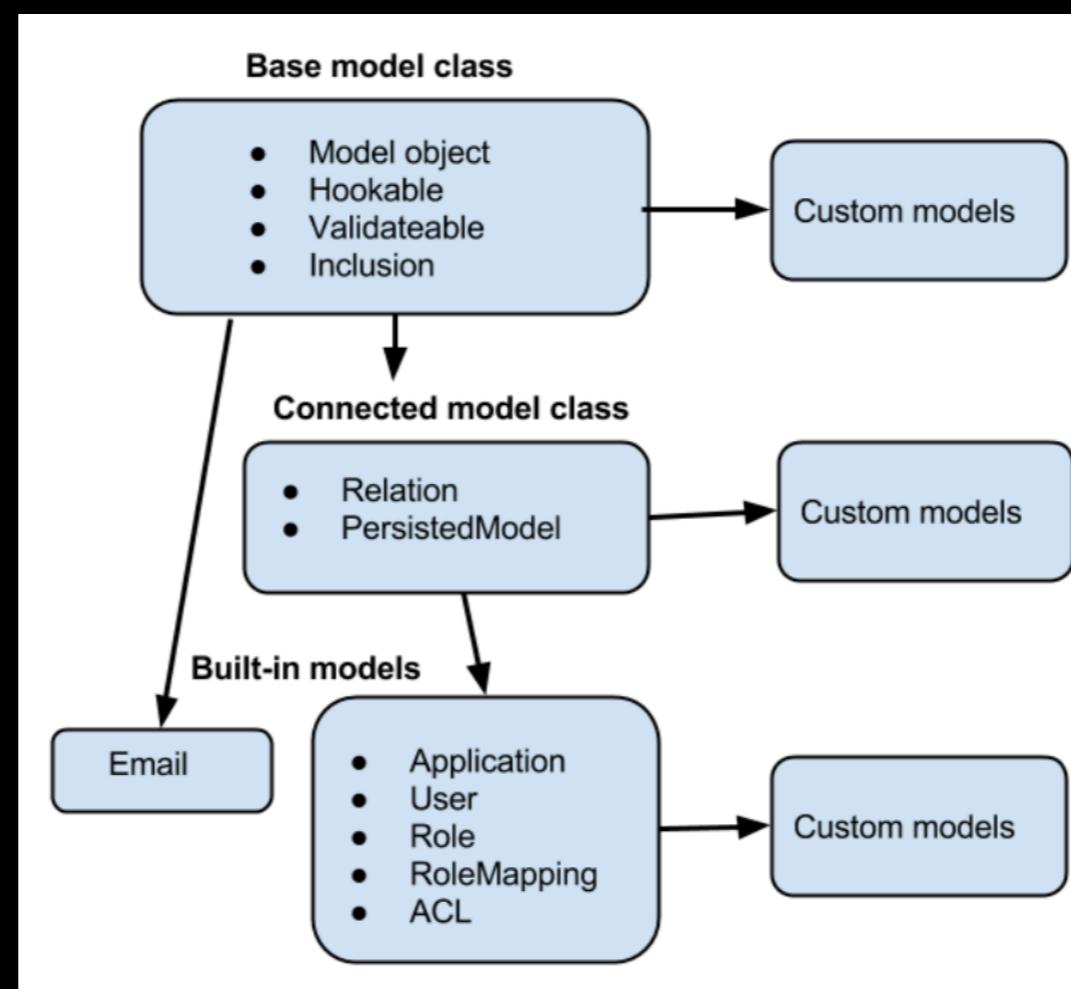
Ещё немного скриншотов

Data sources



Built-in models

Every LoopBack application has a set of predefined [built-in models](#) such as User, Role, and Application, so you don't have to create these common models from scratch.



Creating custom models

Creating model relations

You can define the following relations between models:

- [BelongsTo relations](#)
- [hasOne relations](#)
- [hasMany relations](#)
- [hasManyThrough relations](#)
- [hasAndBelongsToMany relations](#)
- [Polymorphic relations](#)
- [Embedded relations](#) (`embedsOne` and `embedsMany`)

Using validation methods

82

Method	Description
validatesAbsenceOf	Validate absence of one or more specified properties. A model should not include a property to be considered valid; fails when validated field not blank.
validatesExclusionOf	Validate exclusion. Require a property value not be in the specified array.
validatesFormatOf	Validate format. Require a model to include a property that matches the given format.
validatesInclusionOf	Validate inclusion in set. Require a value for property to be in the specified array.
validatesLengthOf	Validate length. Require a property length to be within a specified range. Three kinds of validations: "min," "max," and "is." Default error messages are: <ul style="list-style-type: none">• min: too short• max: too long• is: length is wrong
validatesNumericalityOf	Validate numericality. Requires a value for property to be either an integer or number.
validatesPresenceOf	Validate presence of one or more specified properties. Requires a model to include a property to be considered valid; fails when validated field is blank.
validatesUniquenessOf	Validate uniqueness. Ensure the value of the property is unique for the model. Not available for all connectors. Currently supported with these connectors: <ul style="list-style-type: none">• In Memory• Oracle• MongoDB
validatesDateOf	Validate if a value for a property is a Date. Requires a model value for property to be of type Date.

Creating, updating, and deleting data

[create](#) - creates a new model instance (record).

[upsert](#) - checks if the instance (record) exists, based on the designated [ID property](#), which must have a unique value; if the instance already exists, the method updates that instance. Otherwise, it inserts a new instance.

[findOrCreate](#) - Find one instance matching the filter object provided as the first parameter. If found, returns the object. If not found, creates a new instance (record).

[save](#) - Save model instance. If the instance doesn't have an ID, then calls [create](#) instead. Triggers: validate, save, update, or create.

[updateAll](#) - updates multiple instances (records) that match the specified [where clause](#).

[updateAttribute](#) - Update a single attribute (property).

[updateAttributes](#) - Update set of attributes (properties). Performs validation before updating.

[destroyAll](#) - Delete all model instances that match the optional [Where filter](#).

[destroyById](#) - Delete the model instance with the specified ID.

Querying data

LoopBack supports the following kinds of filters:

- [Fields filter](#)
- [Include filter](#)
- [Limit filter](#)
- [Order filter](#)
- [Skip filter](#)
- [Where filter](#)

Adding logic to models

[Remote methods](#) - REST endpoints mapped to Node functions.

[Remote hooks](#) - Logic that triggers when a remote method is executed (before or after).

[Operation hooks](#) - Logic triggered when a model performs create, read, update, and delete operations against a data source.

Remote methods

СОЗДАНИЕ КАСТОМНОГО МЕТОДА

Remote hooks

СОЗДАНИЕ ХУКА НА КАСТОМНЫЙ МЕТОД

Operation hooks

access

before
save

after save

before
delete

after
delete

loaded

persist

Defining middleware

```
function myMiddlewareFunc([err,] req, res, next) {  
  // ...  
}
```

Name	Type	Optional?	Description
err	Object	Required for error-handling middleware.	Use <i>only</i> for error-handling middleware. Error object, usually an instance of <code>Error</code> ; for more information, see Error object .
req	Object	No	The Express request object .
res	Object	No	The Express response object .
next	Function	No	Call <code>next()</code> after your application logic runs to pass control to the next middleware handler.

COMMAND-LINE-TOOLS

COMMAND-LINE-TOOLS

LB

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

COMMAND-LINE-TOOLS

LB

LB PROPERTY

LB DATASOURCE

LB MODEL

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

LB PROPERTY

LB RELATION

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

LB PROPERTY

LB RELATION

LB ACL

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

LB PROPERTY

LB RELATION

LB ACL

LB REMOTE-METHOD

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

LB MIDDLEWARE

LB PROPERTY

LB RELATION

LB ACL

LB REMOTE-METHOD

COMMAND-LINE-TOOLS

LB

LB DATASOURCE

LB MODEL

LB MIDDLEWARE

LB PROPERTY

LB RELATION

LB ACL

LB REMOTE-METHOD

LB SWAGGER

COMMAND-LINE-TOOLS

loopback4 - это не loopback3

loopback4 - это не loopback3

APPLICATION LOGIC

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

CONTROLLERS

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

CONTROLLERS

REPOSITORIES

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

CONTROLLERS

REPOSITORIES

DECORATORS

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

CONTROLLERS

REPOSITORIES

DECORATORS

OPENAPI

loopback4 - это не loopback3

APPLICATION LOGIC

TYPE SCRIPT

DEPENDENCY INJECTION

CONTROLLERS

REPOSITORIES

DECORATORS

OPENAPI

GRAPHQL

КОНЕЦ





Вопросы / ответы

slak@samaradom.ru

nikolasmelui - соцсети

cto@multikey.studio