

Sails.js Cheat Sheet

Getting Started & CLI

Sails.js is an enterprise-grade MVC framework for Node.js with real-time WebSockets and auto-generated REST APIs.

- **Install CLI:** `npm install -g sails`
- **New Project:** `sails new my-app`
- **Start Server:** `sails lift`
- **Generate API (Model + Controller):** `sails generate api user`
- **Generate Standalone Action:** `sails generate action user/login`
- **Generate Model:** `sails generate model post`
- **Interactive Console:** `sails console`

Routing & Blueprints

Custom Routes (config/routes.js)

```
module.exports.routes = {
  'GET /': { view: 'pages/homepage' },
  'POST /api/v1/users':
    'UserController.create',
  'GET /api/v1/users/:id': 'user/find-one', // Actions2 syntax
  'PUT /api/v1/posts/:id': { action:
    'post/update', cors: true },
};
```

Blueprints API

Auto-generates full REST endpoints for models out-of-the-box.

- `GET /user`: Find all records (with query filtering)
- `GET /user/:id`: Find record by ID
- `POST /user`: Create new record
- `PATCH /user/:id`: Update record
- `DELETE /user/:id`: Destroy record

Waterline ORM & Models

```
api/models/User.js
module.exports = {
  attributes: {
    email: { type: 'string', required:
true, unique: true, isEmail: true },
    password: { type: 'string',
required: true, protect: true },
    age: { type: 'number', min: 0 },
  },
};
```

```
// 1-to-Many relation
posts: { collection: 'post', via:
'author' },
},
};
```

Model Queries

```
• const users = await User.find({ age:
  { '≥': 21 } }).sort('createdAt
  DESC').limit(10);
• const user = await
  User.findOne({ email });
• const
  newUser = await User.create({ email,
  password }).fetch();
• const updated = await
  User.updateOne({ id }).set({ age: 22 });
• await User.destroyOne({ id });
• const authorWithPosts = await
  User.findOne({ id }).populate('posts');
```

Controllers & Actions (Actions2)

```
api/controllers/user/find-one.js
module.exports = {
  friendlyName: 'Find one user',
  description: 'Returns a single user
  profile.',

  inputs: {
    id: { type: 'number', required:
true, description: 'User ID' },
  },

  exits: {
    success: { responseType: 'view',
viewTemplatePath: 'pages/user/
  profile' },
    notFound: { responseType:
'notFound' },
  },
};
```

```
fn: async function (inputs, exits) {
  const user = await
  User.findOne({ id: inputs.id });
  if (!user) { return
  exits.notFound(); }
```

```
return exits.success({ user });
},
};
```

Policies & Access Control

Middleware executed before controllers or actions.

```
// api/policies/isLoggedIn.js
module.exports = async function (req,
res, proceed) {
  if (req.session.userId) {
    return proceed();
  }
  return res.forbidden();
};
```

```
// config/policies.js mapping
module.exports.policies = {
  '*': 'isLoggedIn',
  'user/login': true, // Publicly
  accessible
};
```

Real-time WebSockets (Sails Sockets)

Client Subscription:

```
io.socket.get('/user', (data, jwr) =>
{ console.log(data); });
io.socket.on('user', (event) =>
{ console.log('Change detected:',
event); });
```

Server Broadcast:

```
sails.sockets.broadcast('roomName',
'message', { text: 'Hello' });
```

Helpers

- **Invoke helper:** `await sails.helpers.sendEmail.with({ to, subject, text });`