

Turning the Static Dynamic

Divya Sasidharan





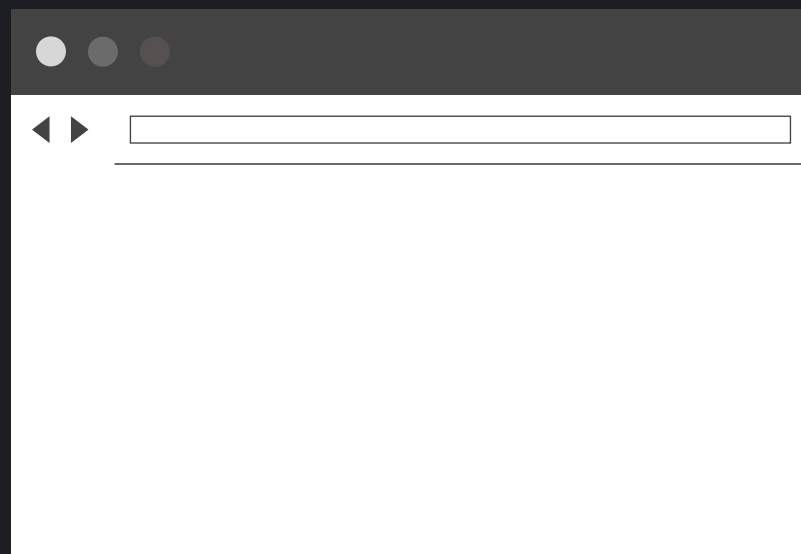
Divya Sasidharan

@shortdiv

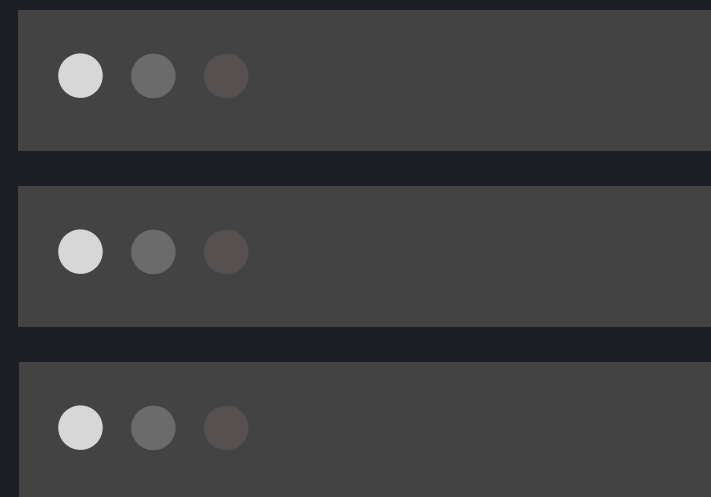
What is “Static”?

Static

Dynamic



VS





secure



fast



flexible



scalable

Being **"Static"**

NEXT.js



NUXT



GitHub Pages



GitBook



React Static

JAMStack



Wes Bos

@wesbos

Following

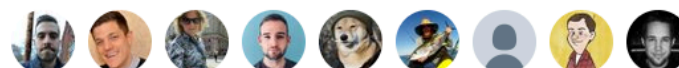


By popular request. The jam, stacked.



2:02 PM - 11 Apr 2019

19 Retweets 394 Likes



10

19



394



JAMStack

JavaScript

API

Markup

Build



Build

Markup

Build

Markup

JavaScript

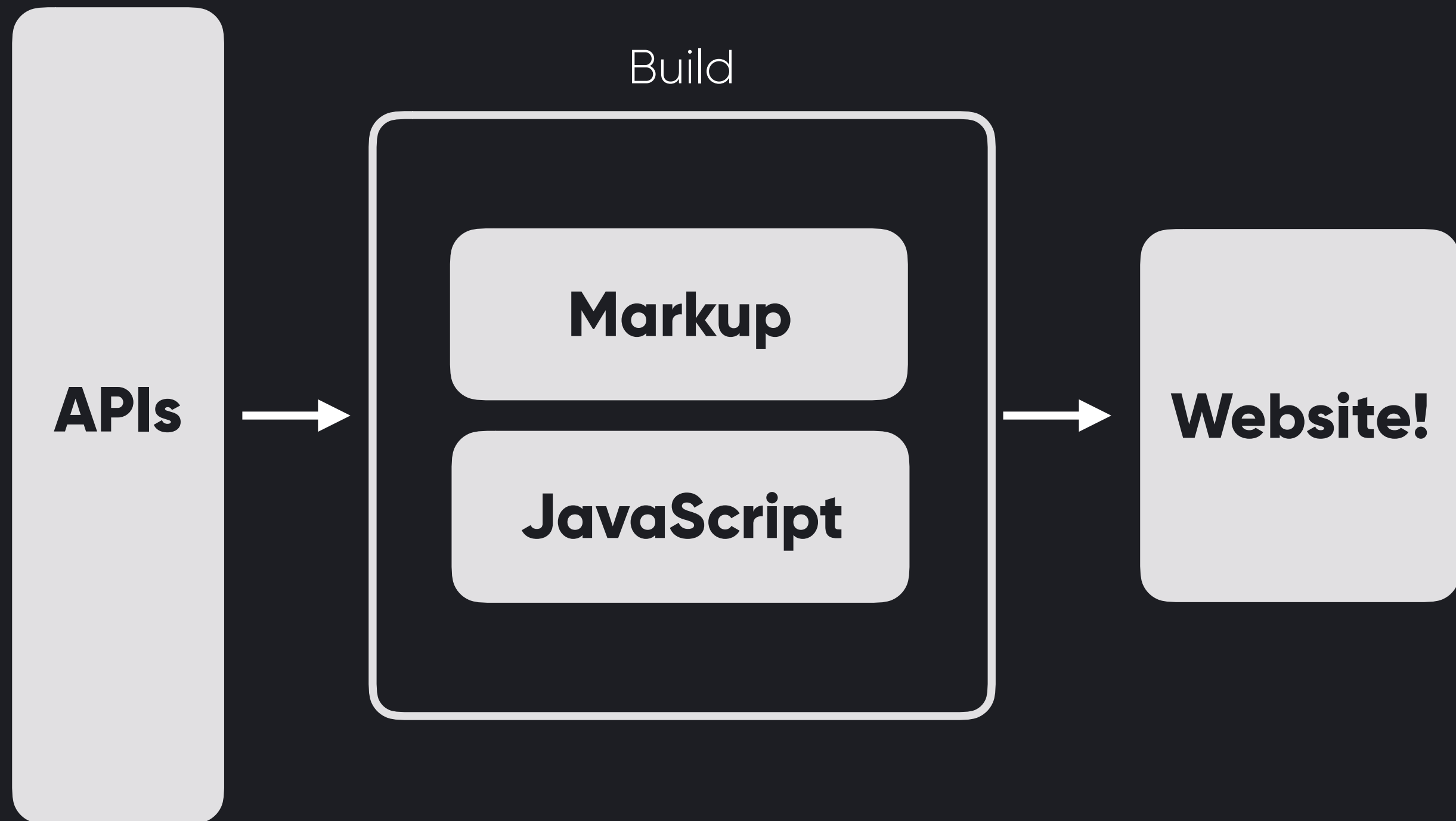
Build

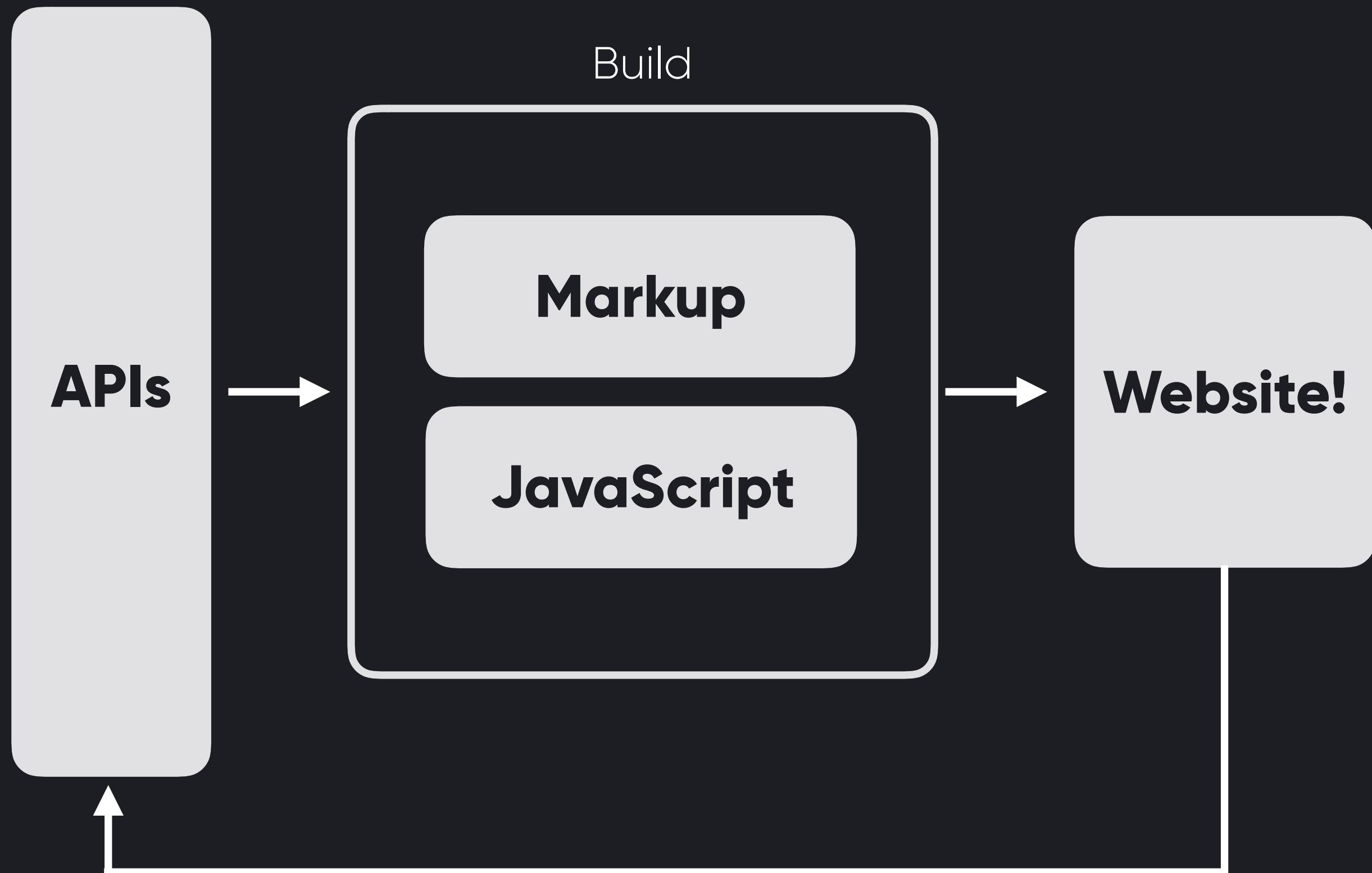
Markup

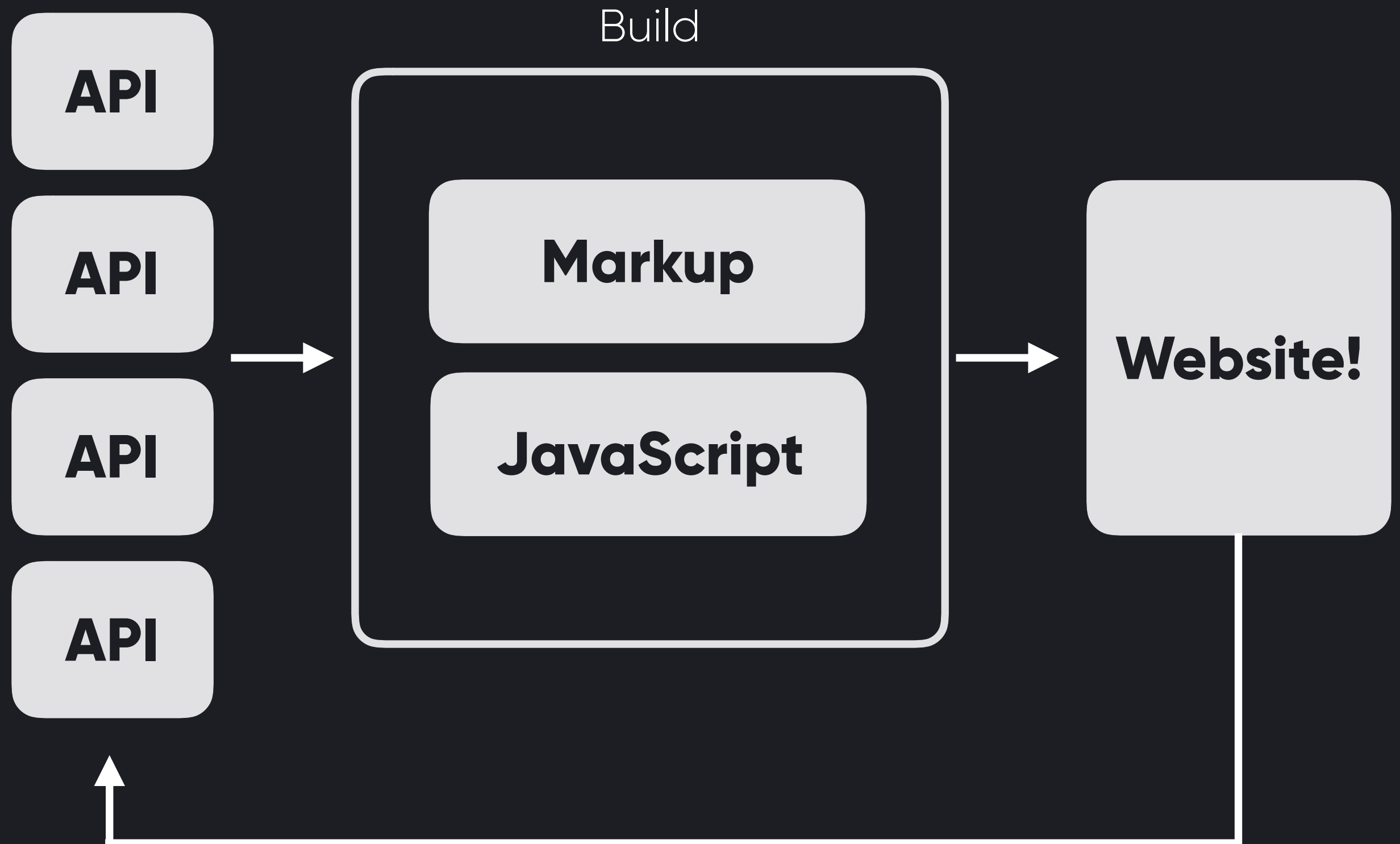
JavaScript



Website!







"Decoupled"

Going “**Dynamic**”

Tools of the Trade

Build Automation

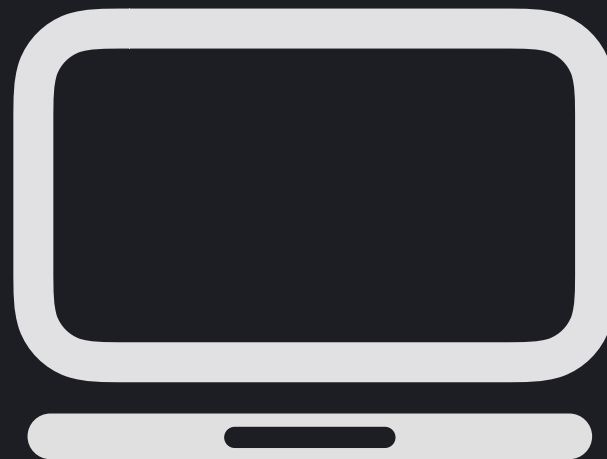
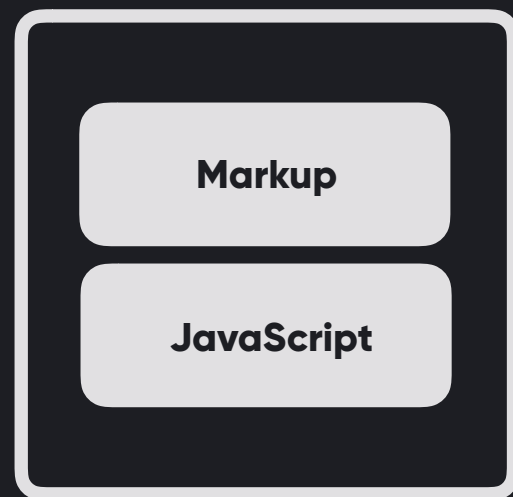
Event Triggers

APIs

Serverless

Build Automation

Build





netlify.toml x



```
[build]
  command="npm run build"
  publish="public"
```

Set your watch by Netlify ([America/New_York](#))
Made by [Phil](#) for some [bizarre reason](#).

3:27 PM

https://setyourwatchby.netlify.com/America/New_York

Event Triggers

if this then that

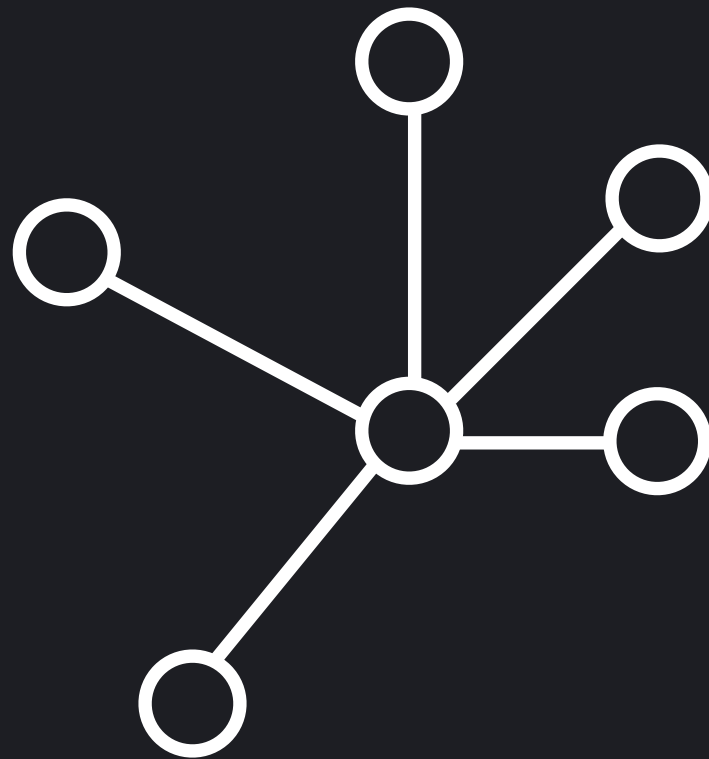
Event Triggers

Form Submissions
Build/Deploy Events
Split Test Events
Auth Events

Form Submissions
Content Updates
Git Activity
Event Hooks

Demo Time!

APIs



ecosystem

\ 'ē-kō-si-stəm

Form Handling

Search

Push Notifications

Authentication

Database

Static Hosting

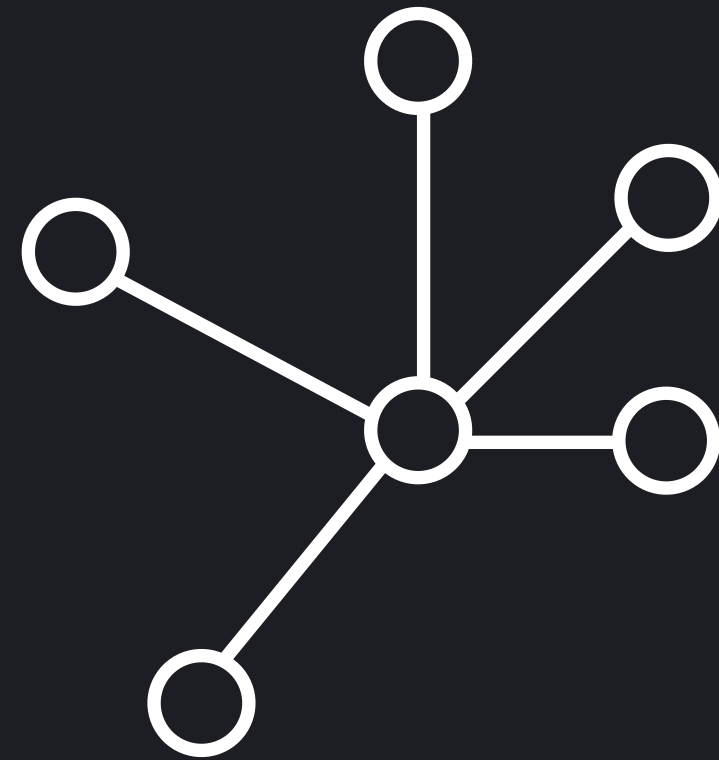
Payments

Media Storage

Headless CMS

Monitoring

FAAS Providers



ecosystem

\ 'ē-kō-si-stəm



The Power of SERVERLESS *for Front-End Developers*



What?



Services



Ideas



Resources

Serverless. It's kind of a phenomenon. Let's get one thing out of the way: it still involves servers, so that word *serverless* might feel a bit disingenuous. It's actually a new way to pay for and work with servers that, in many cases, is *cheaper* and *easier* than buying and managing your own servers.

Web developers of all kinds can take advantage of it, but **this site is particularly interested in how it relates to and helps front-end web developers**. Here's one way to think about it: you can take your front-end skills and do things that typically only a back-end can do. You can write a JavaScript

thepowerofserverless.info

Serverless

Functions *as a* Service



Demo Time!

Demo Time!



Identity

Identity Free

API endpoint: <https://you-auth-to-know.netlify.com/.netlify/identity>

[Learn more in the documentation](#) ↗

⚙ Settings and usage

Identity audit log

🔍 Search users

Invite users



Divya

divya@netlify.com

No role set

Created on Mar 14 (19 days ago)



Div

hello@shortdiv.com

No role set

Created on Feb 11 (2 months ago)




```
exports.handler = function(event, context, callback) {
  const data = JSON.parse(event.body);
  const { user } = data;

  console.log(user.email);
  console.log("identity yourself", context.clientContext.identity);

  const validateUser = email => {
    if (email.split("@")[1] === "netlify.com") {
      return ["editor"];
    } else {
      return ["visitor"];
    }
  };

  const roles = validateUser(user.email);

  const responseBody = {
    app_metadata: {
      roles,
      my_user_info: "this is user info that the user can't change from the UI"
    },
    user_metadata: {
      ...user.user_metadata,
      custom_data_from_function: "hurray this is some extra metadata"
    }
  };

  callback(null, {
    statusCode: 200,
    body: JSON.stringify(responseBody)
  });
};
```

```
exports.handler = function(event, context, callback) {
  const data = JSON.parse(event.body);
  const { user } = data;

  console.log(user.email);
  console.log("identity yourself", context.clientContext.identity);

  const validateUser = email => {
    if (email.split("@")[1] === "netlify.com") {
      return ["editor"];
    } else {
      return ["visitor"];
    }
  };

  const roles = validateUser(user.email);

  const responseBody = {
    app_metadata: {
      roles,
      my_user_info: "this is user info that the user can't change from the UI"
    },
    user_metadata: {
      ...user.user_metadata,
      custom_data_from_function: "hurray this is some extra metadata"
    }
  };

  callback(null, {
    statusCode: 200,
    body: JSON.stringify(responseBody)
  });
};
```

```
exports.handler = function(event, context, callback) {
  const data = JSON.parse(event.body);
  const { user } = data;

  console.log(user.email);
  console.log("identity yourself", context.clientContext.identity);

  const validateUser = email => {
    if (email.split("@")[1] === "netlify.com") {
      return ["editor"];
    } else {
      return ["visitor"];
    }
  };

  const roles = validateUser(user.email);
```

```
const responseBody = {
  app_metadata: {
    roles,
    my_user_info: "this is user info that the user can't change from the UI"
  },
  user_metadata: {
    ...user.user_metadata,
    custom_data_from_function: "hurray this is some extra metadata"
  }
};
```

```
callback(null, {
  statusCode: 200,
  body: JSON.stringify(responseBody)
});
```

```
exports.handler = function(event, context, callback) {
  const data = JSON.parse(event.body);
  const { user } = data;

  console.log(user.email);
  console.log("identity yourself", context.clientContext.identity);

  const validateUser = email => {
    if (email.split("@")[1] === "netlify.com") {
      return ["editor"];
    } else {
      return ["visitor"];
    }
  };

  const roles = validateUser(user.email);

  const responseBody = {
    app_metadata: {
      roles,
      my_user_info: "this is user info that the user can't change from the UI"
    },
    user_metadata: {
      ...user.user_metadata,
      custom_data_from_function: "hurray this is some extra metadata"
    }
  };

  callback(null, {
    statusCode: 200,
    body: JSON.stringify(responseBody)
  });
};
```

```
{
  app_metadata: {
    roles,
    ...user.app_metadata,
    my_user_info: "some user info"
  },
  user_metadata: {
    ...user.user_metadata,
    custom_data_from_function: "hurray extra metadata!"
  }
};
```

```
{
  app_metadata: {
    roles,
    ...user.app_metadata,
    my_user_info: "some user info"
  },
  user_metadata: {
    ...user.user_metadata,
    custom_data_from_function: "hurray extra metadata!"
  }
};
```




JAMstack Boston

Divya Sasidharan



@shortdiv