

RT-MVC

Real Time Model/View/Controller Applications

Daniel Erickson / **qConSF**

**WHAT'S GOING TO
WIN
REAL TIME OR MVC?**

REAL TIME?

YEAH!

'CUZ SOME PEOPLE SAY:

- MVC doesn't scale well
- MVC is dead

BUT MVC IS HERE TO STAY

- Gives teams structure to get things done
- Allows new people to jump into your project
- Easy to get started with

**WHAT'S GOING TO
WIN
REAL TIME OR MVC?**

MVC?

TOTALLY!

'CUZ PEOPLE THINK:

- Real Time is hard
- Maintain a new stack
- Event Machine and Tornado suck
- Front end code is messy

BUT REAL TIME IS ALSO HERE TO STAY

- Use WebSockets to get updates on the front end
- Instant feedback for your users
- Works really well on mobile

**WHAT'S GOING TO
WIN
REAL TIME OR MVC?**

BOTH!

REAL TIME AND MVC ACTUALLY MIX PRETTY WELL

YOU CAN:

- Define your model once and use it all over the place
- use your templates on both sides of HTTP
- Mirror events from the backend to the frontend
 - instance created
 - instace updated
 - instance removed

HOW WE SOLVED THE REAL TIME PROBLEM WITH GEDDY

**WAIT,
WHAT'S GEDDY?**

THE ORIGINAL MVC FRAMEWORK FOR NODE.JS

- Very familiar if you know Rails
- Awesome noSQL and SQL ORM
- Many different templating languages

**AND NOW IT
SUPPORTS REAL TIME
RIGHT OUT OF THE BOX**

**SO LETS MAKE AN MVC APP
THAT SUPPORTS REAL TIME**

GENERATE AN APP

```
$ npm install -g geddy  
$ geddy app -rt demo  
$ cd demo  
$ geddy scaffold -rt thing title description  
$ geddy
```

TEST IT OUT

Open up <http://localhost:4000/things>
and <http://localhost:4000/things/add>

FOR THOSE OF YOU FOLLOWING ALONG AT HOME

1. add a `thing`
2. watch it show up in the other window
3. update the `thing`
4. watch it change in the other window
5. remove the item
6. watch it get removed from the other window.

WHAT JUST HAPPENED?

```
geddy app -rt demo
```

Generated a demo app with realtime enabled

WHAT JUST HAPPENED?

```
geddy scaffold -rt thing ...
```

Generated a `thing` resource

- `thing` model
- made that model available on the front end
- `things` controller
- routes
- views that subscribe to the realtime events

HOW DOES IT WORK?

ON THE SERVER SIDE:

- geddy sees the `realtime` flag in your config file
- copies your models over to a public js file
- sets up listeners for each model's lifecycle events
- proxies those lifecycle events to `socket.io`

HOW DOES IT WORK?

ON THE CLIENT SIDE:

- registers listeners for all model related events
- proxies them to the appropriate models
- scaffolds some code that hooks into real time events

THATS PRETTY COOL, BUT IS IT EASY TO CHANGE?

- modify your model once and you get it everywhere
- emit custom events on your models
- all generated code can be modified

WHAT'S NEXT ON THE ROADMAP?

- Create an AJAX adapter for the front-end ORM
- Real time queries
- Per instance real time events
- A better way to share templates

WANT TO HELP OUT?

CHECK OUT GEDDY AT

<http://geddyjs.org>

Join the mailinglist
open bugs on github
send us pull requests

ANY QUESTIONS?

