

O que é

RTA, Real Time App ou Aplicativo de tempo real, é um programa que funciona/responde dentro de um período de tempo que o usuário percebe como imediato ou atual. A latência deve ser medida em segundos. O uso de RTAs é chamado de computação em tempo real (RTC).

Quem utiliza

- **Documento sendo modificado ao vivo por outra pessoa** (Trello, Dropbox Paper ou Google Docs)
- **Streaming de vídeo** (Netflix, Amazon Play)
- **Streaming de áudio** (Spotify, Soundcloud)
- **Chat de texto** (Facebook, Slack)
- **Sistema de pagamentos on-line** (Paypal) <https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/>
- **Aplicativo móvel** (Linkedin, Walmart, Uber)
- **Dados de trajes espaciais para banco de dados em nuvem** (NASA)
- **Páginas web** (Yahoo, Terra, Globo, Uol)
- **Inúmeros projetos** (Google, Microsoft, Mozilla, Github)

Para que serve

- **Obter troca de respostas simultâneas entre o lado do servidor com o do cliente;**
- **Trabalhar com tarefas leves** (em caso de tarefas pesadas, não irá travar a aplicação, elas serão sempre executadas em background e quando finalizarem, os dados podem ser tratados através dos call-backs das funções);
- **Proporcionar um comportamento parecido com o mundo real:** as coisas vão acontecendo conforme elas devem acontecer. Estamos em 2018, ninguém precisa atualizar a página para descobrir que o valor mudou. Em uma específica situação, o valor deve ser alterado e a pessoa apenas deve receber uma notificação de que o valor foi alterado.

Quais problemas resolveriam

- Melhor controle: saber quem está acessando o que em uma aplicação e tratar isso
- Permitiria criar novos produtos com melhores experiências: hoje existe um ponto de limite na criatividade e experiência

Vantagens

- **Bastante conteúdo online** – Nos últimos anos muitas pessoas vem buscando essa necessidade, existem muitos projetos open-source, plugins e frameworks para ajudar nesse tipo de projetos.
- **Dominar tecnologias atuais**
- **Possibilidade de novos produtos**
- **Linguagem conhecida** – Não será preciso aprender uma linguagem diferente para criar aplicações em tempo real. Hoje, é possível criar com as linguagens que a empresa já utiliza, php e javascript. Esse tópico não fala dessas diferenças.

Desvantagens

- **Levará tempo** – Como nunca usamos antes, terá um tempo de adaptação para estudar melhor como vai funcionar. Não tem escapatória para aprender qualquer coisa nova.
- **Provavelmente terá que mudar a arquitetura** - Estamos amadurecendo, o mundo e os projetos que criamos estão diferentes. Provavelmente não terá que utilizar o MVC, pois ele foi criado para resolver um problema organizacional, facilitar a manutenção em projetos. Com projetos com um propósito diferente, como real-time, é natural que outros problemas mais complexos apareçam, e é necessário outras arquiteturas para resolver esses problemas.

Por que não devemos adotar

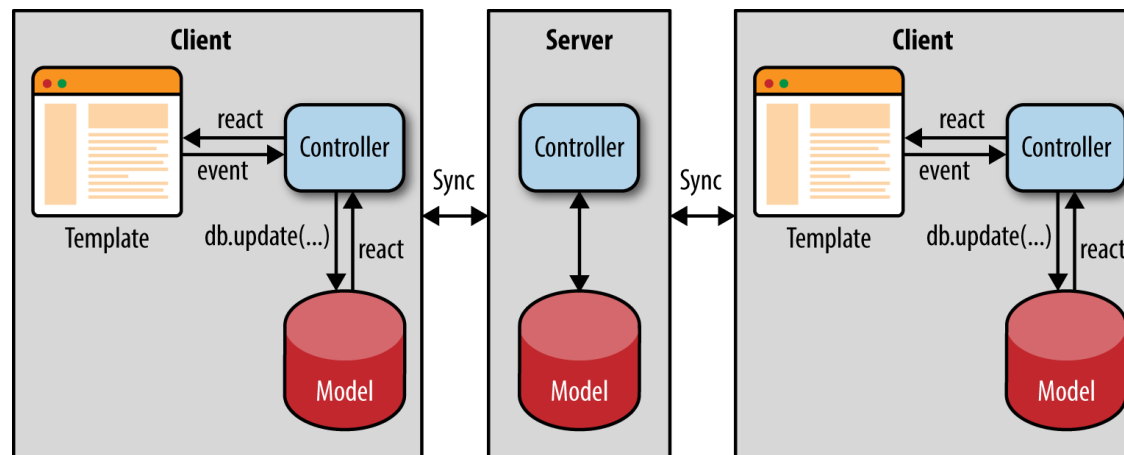
- **Só temos um programador** – Acredito que precisaríamos ter mais pessoas disponíveis para desenvolver e estudar. É necessário de tempo para estudar. Uma pessoa fica realmente complicado.
- **Surgimento dos aplicativos web progressivos** - Mais importante estudar um PWA do que uma aplicação em real-time. Não é uma nova tecnologia, é um conjunto de tecnologias que permitem uma nova forma de conceber um aplicativo para os usuários móveis. O real-time poderia ser aplicado dentro de um PWA.

Onde seria aplicado

Deve-se projetar e implementar maneiras de **enviar dados do servidor para o cliente para mostrar as alterações de outros usuários conforme elas acontecem**.

Também precisa existir uma maneira de **atualizar a tela de maneira reativa em cada cliente, quando esses dados vierem do servidor**. A maioria dos aplicativos em tempo real tem peças funcionais semelhantes.

Normalmente tem mecanismos para **assistir a um banco de dados em busca de alterações**.



Podemos ver como os dados fluem da interação do usuário com a visão. Também podemos ver como as alterações de outros clientes se propagam para todos os usuários e como a visualização é renderizada quando as alterações de dados são enviadas do servidor. Existem três conceitos isomórficos interessantes que surgem nesse tipo de arquitetura: APIs isomórficas, sincronização bidirecional de dados e simulação de cliente no servidor.

Conclusão

- RTA são muito úteis para criar excelentes projetos;
- Os principais/melhores produtos da web utilizam;
- É possível rodar com baixíssimo custo em hardware;
- Boas arquiteturas diminuem custos com hardwares e infraestrutura;
- É possível desenvolver RTA tanto em PHP quanto em Javascript;
- É preciso estudar as arquiteturas para projetos em RTA antes de tomar qualquer decisão;