

webR, et le futur des apps web avec R

Colin Fay, ThinkR
colin@thinkr.fr

Résumé (max 300 mots)

L'un des grands plaisirs de l'ingénierie logicielle est que les choses bougent sans arrêt. De nouvelles technologies, de nouveaux langages, de nouveaux frameworks... De temps en temps, parmi ces nouveautés, de nouvelles choses émergent et changent la manière dont nous construisons des produits.

Au cours des dernières années dans le monde de R, nous avons construit et déployé des applications web et des API de manière assez stable : en construisant des applications {shiny} avec des frameworks comme {golem} ou {rhino}, des API avec {plumber}, pour ensuite les envoyer sur un serveur qui peut lancer R et rendre notre code disponible au monde. Ces derniers mois, une nouvelle technologie est apparue dans le "R World" : webR, une version de R compilée pour WebAssembly (WASM), permettant d'exécuter R directement dans le navigateur ou à l'intérieur de NodeJS, sans avoir besoin d'une installation R. Ce qui ouvre un champs des possibles incroyable pour le développement web, JavaScript étant l'outil de choix pour la construction d'applications et d'API.

Dans cette présentation, Colin commencera par expliquer ce qu'est webR, projet développé par l'équipe Posit, et comment cela changera notre façon de penser la construction et le déploiement de code R pour le web.

Il présentera ensuite deux packages NodeJS expérimentaux sur lequel il travaille : webrcli, interface en ligne de commande offrant un "kit de démarrage" pour des projets avec NodeJS / webR, et spidy, une boîte à outils pour interagir avec webR depuis une application NodeJS.

Et enfin, Colin se concentrera également sur les défis qui se poseront avec webR, et comment nous construirons des applications web avec R à l'avenir.

Mots-clefs (3 à 5) : webr, API, shiny, développement web

Développement

L'un des aspects les plus exaltants du métier d'ingénieur logiciel réside dans sa nature perpétuellement changeante. Dans cet univers en constante évolution, de nouvelles technologies, de nouveaux langages et de nouveaux frameworks émergent régulièrement, apportant avec eux des perspectives nouvelles et des possibilités inédites pour la conception et le développement de produits et de service. R n'est pas étranger à cette constante évolution, et s'aventure d'année en année sur des terrains nouveaux, tant pour l'analyse de données que pour la construction d'outils rendant accessible au monde extérieur les résultats de nos travaux : on pense ici à {shiny} bien sûr, mais aussi à {plumber}, {markdown}, quarto... tant de technologies appelées depuis R et qui permettent de servir sur le web du code directement sur le web, plutôt que dans un terminal.

Sur le web, l'information se consomme de deux grandes manières : via des applications web, ou via des systèmes d'API REST. Ces dernières années, nous avons pu constater une certaine stabilité dans la manière dont nous construisons et déployons ces applications et services :

- Pour des apps web, avec des outils bien établis tels que {shiny}, en association avec des frameworks comme {golem} ou {rhino}. Avec eux, nous avons pu créer des applications interactives et des interfaces utilisateur dynamiques
- Des solutions telles que {plumber} nous ont permis de développer des API robustes et performantes.

Ces applications étaient ensuite déployées sur des serveurs capables de lancer du code R, rendant ainsi le code R accessible à un large public, via des services comme Posit Connect, des technologies comme Docker, et d'autres.

Et si nous étions aujourd'hui à l'aube d'un nouveau bouleversement ?

Car oui, même dans ce paysage relativement stable, des bouleversements se produisent. Ces derniers mois ont vu l'émergence d'une technologie qui devrait révolutionner la façon dont nous faisons du web avec R : webR, outil développé par l'équipe open source de chez Posit. Un terme mystérieux qui désigne la compilation de R pour WebAssembly (WASM), un outil destiné à rendre possible l'exécution de code qui était jusqu'ici traditionnellement exécuté sur le serveur, directement dans les navigateurs web modernes. Une exécution également possible au sein de l'environnement NodeJS, le runtime JavaScript sur serveur.

Cette innovation ouvre des perspectives incroyables pour le développement web, en permettant l'exécution de code R directement dans le navigateur, sans nécessiter une installation préalable de R sur l'appareil de l'utilisateur. Avec JavaScript qui continue d'être le langage de programmation dominant pour la construction d'applications web et d'API, cette convergence entre R et WebAssembly promet de révolutionner la façon dont nous pensons et concevons les applications web basées sur R.

Dans le cadre de cette présentation, Colin nous guidera à travers les méandres de webR, nous offrant un aperçu de cette technologie émergente et de son impact sur le développement web avec R. Il commencera par expliquer en quoi consiste exactement webR, en mettant en lumière ses avantages et ses implications pour notre manière de créer et de déployer des applications web.

Colin nous présentera deux outils expérimentaux sur lesquels il est en train de travailler : webrcli, interface en ligne de commande permettant de faciliter la création de projets avec NodeJS / webR, et spidyr, une boîte à outils pour interagir avec webR depuis une application écrite en NodeJS. Ces outils permettent de créer des applications NodeJS capables d'appeler du code R via webR, offrant ainsi une solution élégante pour intégrer R dans des environnements JavaScript et tirer parti des capacités avancées de cette combinaison de technologies.

Imaginez un instant : la puissance et la simplicité de la manipulation de données dans R, combinée à l'incommensurable paysage d'outils pour le web de NodeJS — le tout, dans le but inavoué de réconcilier les équipes de développeurs web et les équipes de développeurs R, et de leur donner un cadre de travail commun.

Enfin, Colin abordera les défis potentiels associés à l'adoption de webR, notamment en ce qui concerne la gestion des performances, la sécurité et la maintenance à long terme. Il partagera sa vision sur la manière dont nous pourrions surmonter ces défis et construire des applications web robustes et évolutives avec R et JavaScript à l'avenir.

Références :

- <https://webr.r-wasm.org/>
- <https://github.com/ColinFay/webcli>
- <https://github.com/ColinFay/spidyr>
- <https://colinfay.me/webcli-and-spidyr/>
- <https://github.com/ColinFay/webrspongebob>