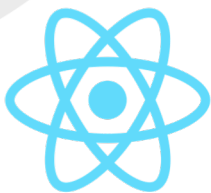




Construisons votre **j**SOLUTION



L'ENVIRONNEMENT **IDEAL POUR EXTJS**



PARIS – 28.04.2018

Vincent Munier

Manager Technique





Pourquoi cette présentation ?

LE PLANNING

PLANNING

- Développement
 - Sencha CMD, Éditeurs de code, Outils intégrés
 - Système de contrôle de versions
 - Debugging, testing
 - Déploiement local et accès aux données
- Intégration
 - Outil d'intégration continue
 - Testing et building
- Déploiement



DEVELOPPEMENT

DEVELOPPEMENT

SENCHA CMD

Indispensable au développement d'une application Ext JS 6

- Génération d'application et workspace
- Génération de vues, modèles et packages
- sencha app watch utile en phase de theming
- Upgrade de framework facilité
- sencha switch pour gérer les différentes versions de la CMD
- Build profiles pour générer plusieurs types de builds (production, testing, thèmes différents...)

EDITEURS DE CODE

Visual Studio Code, Suite JetBrains

- Visual Studio Code quand maîtrise unique du front-end
- Suite JetBrains (IntelliJ par ex.) sur un projet Java Back / JS Front
- Plugins comme "Beautify" ou "ESLint" pour harmoniser les normes de codage
- Plugin Sencha efficace, beaucoup de temps gagné
 - Autocomplétion, snippets, ctrl+click, doc rapide
 - Plus abouti sur les outils JetBrains

DEVELOPPEMENT

OUTILS INTÉGRÉS

Sencha Architect

- Pratique pour du prototypage de vues
- Peut servir à harmoniser visuellement les layouts
- Peut aider les débutants à monter en compétences sur les layouts et différents composants des vues Ext JS

DEVELOPPEMENT

OUTILS INTÉGRÉS

Sencha Themer

- Un développeur n'a pas forcément la fibre artistique
- Outil ergonomique, grosses économies de temps passé sur le style
- Possibilité de générer un package intégrable dans plusieurs applications
- Possibilité de thémer une application à chaud

Système de contrôle des versions : Git

- Facilement intégrable aux différents outils d'intégration continue
- La norme dans l'open-source
- Quelques best-practices:
 - Une branche par fonctionnalité ou par bug, séparées dans des dossiers FEATURE ou BUG
 - Corrélation entre noms de branches et tâches planifiées
 - Tagger chaque version après merge dans la branche principale

DEVELOPPEMENT **DEBUGGING**

Plusieurs outils

- Console Chrome
 - La plus performante
 - Nombreuses fonctionnalités :
 - Beautification
 - Interprétation transparente des source-maps
 - Breakpoints conditionnels
 - Callstack asynchrone
 - Pseudo-émulation de devices mobiles
 - Debugging distant depuis mobile ou tablette

DEVELOPPEMENT **DEBUGGING**

Plusieurs outils

- Sencha Inspector
 - Vue d'ensemble et inspection individuelle de composants Ext (pas uniquement du DOM généré)
 - Affichage des configs et fonctions des composants, informations d'héritage ou de surcharge
 - Vue d'ensemble des données utilisées par les composants (stores, modèles etc...)
 - Logging de tous les événements déclenchés d'une application

DEVELOPPEMENT **DEBUGGING**

Plusieurs outils

- Bryntum RootCause
 - API JS intégrable à n'importe quelle application Web
 - Remontée des erreurs JS avec contexte, nombreuses informations sur le navigateur utilisé, stacktrace complète, screenshot etc...
 - Enregistrement d'une vidéo des actions de l'utilisateur avant le déclenchement de l'erreur
 - Gestion des source-maps, mode offline

<https://therootcause.io/>

DEVELOPPEMENT **TESTING**

Sencha Test Studio

- Application standalone de création et d'exécution de tests pour Ext JS
 - Tests unitaires
 - Tests UI & Tests d'intégration (End to End)
- Recorder d'événements pour faciliter la création de tests applicatifs
- Tests asynchrones facilités par une API dédiée
- Tests cross-browser avec intégration à SauceLabs ou Browserstack

DÉPLOIEMENT LOCAL ET ACCÈS AUX DONNÉES

Déploiement local

- Docker, docker-compose
 - Système de création de conteneurs logiciels
 - Permet d'harmoniser simplement les environnements de développement, spécialement côté BackEnd, entre les développeurs
 - Permet le partage de valeurs entre différents OS, versions d'OS etc...

DÉPLOIEMENT LOCAL ET ACCÈS AUX DONNÉES

Accès aux données

- Simulation d'échanges client/server ou implémentation rapide d'API REST
 - Mock-server
 - Node-Red
 - Headless CMS (Gentics Mesh, KeystoneJS...)
 - Simlets Ext JS
 - <https://mockaroo.com>



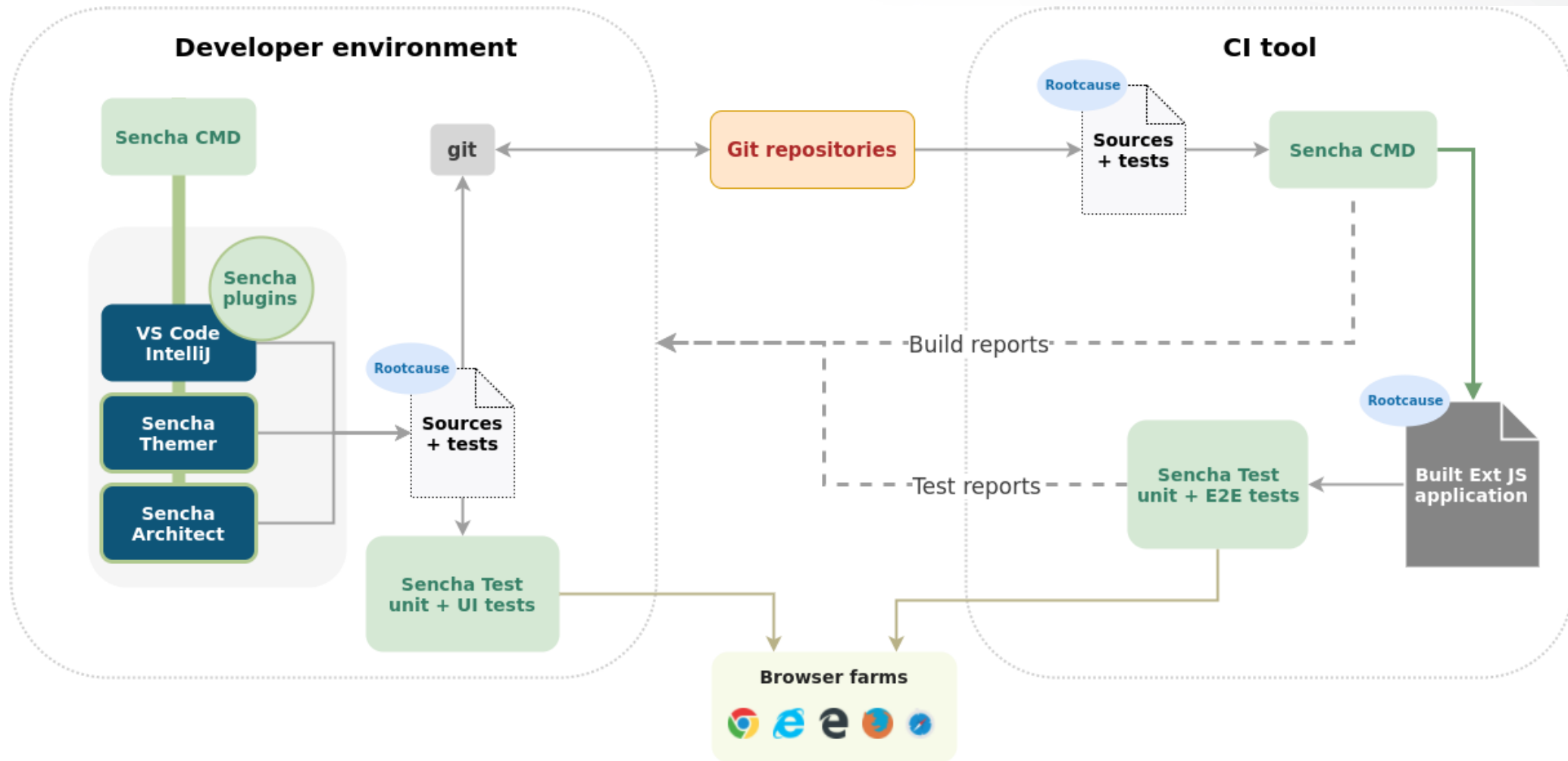
INTEGRATION

TESTING ET BUILDING

Outil d'intégration continue

- Jenkins, TeamCity, TravisCI
 - Sencha CMD facilement déployable sur des serveurs d'IC
 - Sencha Test permet l'exécution de tests en mode headless (sans interface graphique)
 - Tests unitaires
 - Tests applicatifs et UIs via les fermes de navigateur (BrowserStack, SauceLabs)

Vue d'ensemble de l'environnement de développement et d'intégration



DEPLOIEMENT

DEPLOIEMENT

- **Différentes possibilités:**
 - Conteneurs logiciels:
 - Déploiement de conteneurs Docker notamment avec Google Cloud, AWS
 - Kubernetes (gestion de multiples conteneurs, mise à l'échelle en fonction des besoins en ressources, gestion de reprise sur incident)
 - Serverless: Google Cloud Functions, AWS Lambda
 - Déploiement plus traditionnel
 - Serveurs applicatifs PHP, Java, NodeJS



Conclusion

A stylized illustration of a workspace. In the top left, there's a green plant. In the top right, a pencil and a ruler are on a piece of paper. In the bottom left, a white coffee cup and a pair of black-rimmed glasses are visible. In the bottom right, there's a smartphone and a piece of paper with a hand-drawn diagram. The background is a light gray geometric pattern.

Questions ?



Merci de votre attention !

Vincent MUNIER



www.jnesis.com