

السلام عليكم معاكم اسلام محمد و النهاردة هنتكلم عن حاجة مهمة جداً في عالم تطوير الويب وهي Vite : و-Create React-App. وكمان هنعرف الفرق بين WebPack وESBuild ، ونتعرف ليه Vite ممكن تكون اختيار أفضل ليك. وكمان هنتكلم شوية عن إزاي الـ Bundlers زي الـ EsBuild و الـ WebPack بتشتغل عشان تجمع ملفاتك بطريقة يفهمها المتصفح.

الجزء الأول: الفرق بين Vite و Create-React-App

أول حاجة نبدأ بـ: Create-React-App (CRA)

- دي أداة تم تطويرها من React عشان تساعدك تبدأ أي مشروع بسرعة وسهولة. بتديك بيئة تطوير جاهزة. لكن فيها شوية عيوب بسبب استخدام WebPack المبنية بلغة الجافا سكريبت و هنتكلم عنها بالتفصيل كمان شوية

طيب، إيه بقى Vite ؟

Vite اداة جديدة وسريعة جداً بتديك تجربة تطوير ممتازة بفضل سرعة التحميل والتحديث الكبيرة اللي بتوفرها ودة بالتالي بيخلي عملية التطوير اسهل بكثير ودة بفضل استخدام الـ ESBuild المبنية بلغة GO

ودي خيلينا ننتقل للجزء التالي وهو

الـ Bundlers زي WebPack و ESBuild: إزاي بتشتغل؟

الـ Bundlers زي WebPack و ESBuild بتقوم بتجميع وتحويل الملفات بشكل يخليها تتوافق مع المتصفح.

ف إنت لما تكتب كود JavaScript ، CSS ، أو تستخدم صور ، البندلر بيجمع الملفات دي ويحولها إلى شكل واحد أو مجموعة من الملفات اللي المتصفح بيحتاجها عشان يشغل الموقع. عملية التجميع دي مهمة عشان المتصفح يقدر يفهم ويعرض محتوى موقعك بشكل صحيح.

- WebPack: بيجمع الملفات ويحطها في شكل منظم وسهل التحميل، لكن العملية دي بتأخذ وقت طويل.
- ESBuild: بيعمل نفس العملية بس أسرع بكثير بفضل تقنيات اللغة المكتوب بيها.

ودلوقتي هنشرح بشكل مفصل سبب سرعة Vite مقارنة بـ Create-React-App

لما تستخدم CRA ، الـ WebPack بيقيم بعملية بناء وتجميع ملفات المشروع بتاعتك، وده بييشمل 3 مراحل وهما الـ Bundling والـ Transpiling والـ Optimization :

- **التجميع (Bundling):** WebPack بيجمع ملفات الجافا سكريبت، CSS ، والصور، ويحولها لملفات نهائية يمكن للمتصفح فهمها.
- **التحويل (Transpiling):** WebPack بيحول الكود من صيغة حديثة (زي ES6) لصيغة تتوافق مع المتصفحات القديمة.
- **التحسين (Optimization):** WebPack بيحسن الملفات ويقلل حجمها عشان يزود سرعة تحميل الموقع.
 - لكن في مرحلة التطوير، كل ما تعمل تعديل في الكود:
 - WebPack محتاج يعيد بناء المشروع بالكامل، حتى لو كان التغيير بسيط. وعملية إعادة البناء دي بتشمل كل المراحل اللي اتكلمنا عليها

2. طيب إزاي Vite بتشتغل في عملية التطوير:

- Vite بتتعامل مع الموضوع بطريقة مختلفة بفضل الأساليب اللي بتعتمد عليها اول حاجة:

- **Serving Files Directly:** في Vite ، أثناء عملية التطوير ، الملفات تنتقد للمتصفح بشكل مباشر من النظام بتاعك، يعني مش محتاج تمر بمراحل إضافية قبل ما المتصفح يستقبل الملفات. لما بتفتح مشروعك وتعدل على ملف، Vite مش بترجع تبني المشروع كله من الأول. بدل كده، هو بيتعامل مع الملف اللي اتعدل بس مباشرة. مثلاً، لو أنت شغال على صفحة HTML وعدلت نص فيها، Vite هيتعرف على التغيير ده ويعرضه مباشرة على المتصفح بدون ما يعيد بناء باقي الملفات. ده بيوفر وقت كبير وبيخلي عملية التطوير أسرع لأنك بتشوف التغييرات بشكل فوري.
 - **ثاني حاجة تحديثات سريعة:** لما تستخدم Vite ، هو بيعمل تحديث سريع جداً للملفات اللي انت غيرتها بس. يعني، لو عدلت ملف CSS أو JavaScript ، Vite مش بيدمج التغيير مع باقي الملفات ولا يعيد بناء المشروع بالكامل، لكنه فقط يرفع التغيير الجديد للمتصفح.
 - ده معناه إنك هتشوف التغييرات على المتصفح بسرعة جداً، وده بيخلي تجربة التطوير سلسة. مثلاً، لو غيرت لون في ملف CSS، هتشوف التغيير فوراً على الصفحة بدون تأخير، لأن Vite ببساعدك في تطبيق التعديلات بسرعة.
3. وزي ما قولنا ان Vite بتستخدم ESBuild وإلنها مبنية بلغة GO فهي بالتالي مصممة عشان تكون سريعة جداً في تحويل وتجميع الملفات بفضل المميزات المتوفرة زي
 4. **ESBuild :Transformation Speed** بتحول الكود بشكل أسرع بكثير من WebPack ، لأن لغة Go اللي مكتوبة بيها مصممة للسرعة في المعالجة.
 5. ☐ **ESBuild :Parallel Processing** قادرة على التعامل مع تحويل أكثر من ملف في نفس الوقت (بمعنى إنها تقدر تعالج ملفات متعددة في وقت واحد)، وده بيوفر وقت كبير جداً مقارنة بـ WebPack اللي بيشتغل على الملفات ملف ملف .

أتمنى يكون الشرح واضح ومفيد واشوفكم في فيديو جديد والسلام عليكم.