RESUMO DA AULA

17 Introdução ao gerenciamento de pacotes com NPM

Nesta aula, vamos aprender sobre gerenciamento de pacotes no JavaScript, focando nas ferramentas mais utilizadas: NPM (Node Package Manager) e Yarn. Durante o curso, vamos trabalhar principalmente com o NPM, mas importante saber o que o Yarn e como ele se relaciona com o NPM, caso voc se depare com ele durante a sua carreira. O que um Gerenciador de Pacotes? Um gerenciador de pacotes uma ferramenta que facilita o gerenciamento de bibliotecas (pacotes) e suas depend ncias. Pacotes s o trechos de c digo prontos para resolver problemas espec ficos e ajudam a reduzir o tempo de desenvolvimento ao permitir a reutiliza o de funcionalidades j testadas e otimizadas por outros desenvolvedores. Exemplo Pr tico: Imagine que voc precisa de uma biblioteca para manipular datas no seu projeto. Ao inv s de implementar isso do zero, voc pode utilizar pacotes como o Moment.js ou o date-fns. Basta instalar essas bibliotecas no seu projeto usando um gerenciador de pacotes e elas estar o prontas para uso. Esses pacotes, por sua vez, tamb m podem depender de outras bibliotecas para funcionar corretamente. Esse o conceito de depend ncias de pacotes. Um gerenciador como o NPM facilita esse processo, cuidando de todo o encadeamento e instalando todas as depend ncias necess rias automaticamente. Principais Gerenciadores de Pacotes no Ecossistema JavaScript: NPM (Node Package Manager): Desenvolvido pela npm, Inc., que hoje uma subsidi ria do GitHub. o gerenciador de pacotes oficial do Node.js e o mais popular. Yarn: Criado pelo Facebook para melhorar a performance e seguran a do NPM, mas com o passar do tempo ambos evolu ram e se tornaram muito semelhantes em termos de funcionalidades. Ambos os gerenciadores permitem: Instalar pacotes e suas depend ncias. Gerenciar vers es dos pacotes. Remover e atualizar pacotes. Manter um controle centralizado das bibliotecas utilizadas no projeto. Como instalar o NPM? O NPM j vem instalado automaticamente com o Node.js. Ent o, se voc j instalou o Node no seu ambiente, voc j tem o NPM pronto para uso. Verificando a instala o: Para verificar se o NPM est

instalado, abra o terminal (pode ser o Git Bash, CMD no Windows ou terminal do VS Code) e execute: npm --version Se o NPM estiver instalado corretamente, esse comando retornar a vers o instalada. Instalando o Node.js e o NPM (caso ainda n o esteja instalado): Acesse https://nodejs.org/pt-br/download/. Fa a o download do instalador correspondente ao seu sistema operacional. Execute o instalador e siga as instru es na tela. Por padr o, ele instalar o Node.js e o NPM juntos. Ap s a instala o, feche e abra o terminal novamente e verifique a vers o com o comando: npm --version Isso deve confirmar que o NPM foi instalado corretamente. Como instalar o Yarn? Embora o curso se concentre no uso do NPM, til saber como instalar o Yarn, caso seja necess rio. Instalando o Yarn: Acesse https://classic.yarnpkg.com/lang/en/docs/install/ para obter as instru es de instala o para o seu sistema operacional. O Yarn possui um instalador pr prio, mas tamb m pode ser instalado usando o NPM: npm install -g yarn Esse comando instala o Yarn globalmente no sistema. Verificando a instala o: Ap s instalar o Yarn, verifique a vers o: yarn --version O que o arquivo package.json? O package.json o arquivo de configura o padr o para gerenciar pacotes e depend ncias em projetos Node.js. Ele escrito no formato JSON e cont m informa es sobre o projeto, como: Nome do projeto. Vers o. Lista de depend ncias. Scripts para automatizar tarefas. Outras configura es. Esse arquivo criado automaticamente quando usamos o comando npm init dentro de um diret rio de projeto: npm init Durante a execu o desse comando, voc ser guiado por uma s rie de perguntas, como nome do projeto, vers o e descri o. Ao final, ser gerado o package.json com as informa es fornecidas. Instalando Pacotes com NPM Para instalar um pacote no projeto, utilizamos o comando: npm install <nome-do-pacote> Por exemplo, para instalar a biblioteca lodash, basta rodar: npm install lodash Isso adicionar o lodash ao node_modules do projeto e atualizar o package.json com a depend ncia. Importante: Flags de instala o --save: Adiciona o pacote lista de depend ncias (opcional em vers es mais recentes do NPM). --save-dev: Adiciona o pacote como depend ncia de desenvolvimento. Exemplo: npm install jest --save-dev Nesse caso, o pacote jest ser listado apenas como uma depend ncia para o ambiente de desenvolvimento. Conclus o O NPM e o Yarn s o ferramentas essenciais no desenvolvimento de aplica es modernas com JavaScript. Eles nos permitem gerenciar pacotes e

depend ncias de maneira eficiente e simplificada. Durante o curso, vamos usar o NPM para instalar, atualizar e gerenciar pacotes nos nossos projetos. Mas agora voc j tem uma vis o clara do que s o e como utiliz -los no desenvolvimento de aplica es JavaScript.
