Deno DeployDeno Deploy

<https://deno.com/deploy/docs>

目录

P1 1 Guide

P1 1.1 Hello World

P2 1.2 Using JSX

P2 1.3 Serving static assets

P3 1.4 Running scripts locally

P4 2 Tutorials

P4 2.1 Connecting to Postgres

P4 2.1.1 Overview

P4 2.1.2 Setup Postgres

P5 2.1.3 Write and deploy the application

P7 2.2 Discord Slash Command

P7 2.2.1 Create an application on Discord Developer Portal

P8 2.2.2 Create and deploy the hello world Slash Command

P10 2.2.3 Install the Slash Command on your Discord server

P10 2.3 Persist data using FaunaDB

P10 2.3.1 Overview

P11 2.3.2 Build the API Endpoints

P12 2.3.3 Use FaunaDB for Persistence

P16 2.3.4 Deploy the API

P17 2.4 Persist data using DynamoDB

P17 2.4.1 Overview

P18 2.4.2 Setup DynamoDB

P18 2.4.3 Create a Project in Deno Deploy

P18 2.4.4 Write the Application

P20 2.4.5 Deploy the Application

P21 2.5 Persist data using Firebase

P21 2.5.1 Overview

P22 2.5.2 Concepts

P23 2.5.3 Setup Firebase

P23 2.5.4 Write the application

P25 2.5.5 Create a Project in Deno Deploy

P26 2.5.6 Deploy the application

P27 3 Examples

P27 3.1 Examples Gallery

P27 3.1.1 Hello World

P27 3.1.2 Respond with JSON

P27 3.1.3 Redirects

P27 3.1.4 Get client IP address

P27 3.1.5 Handling <form> submissions

P28 3.1.6 Proxying to other servers

P28 3.1.7 Server side rendering with JSX

P28 3.1.8 Wildcard Domain

P29 4 Platform

P29 4.1 Deployments

P29 4.1.1 Creating Deployments

P29 4.1.2 Production vs Preview Deployments

P29 4.1.3 Logs

P30 4.1.4 Crash Reports

P30 4.2 deployctl

P30 4.2.1 deployctl CLI

P30 4.2.2 Usages

P30 4.2.3 deployctl GitHub Action

P31 4.3 Projects

P31 4.3.1 Creating a Project

P31 4.3.2 Settings

P31 4.3.3 Domains

P32 4.3.4 Environment Variables

P32 4.3.5 Git Integration

P33 4.4 Playgrounds

P33 4.4.1 Creating a playground

P33 4.4.2 Using the playground editor

P33 4.4.3 Making a playground public

P33 4.4.4 Exporting a playground to GitHub

P34 4.5 Organizations

P34 4.5.1 Create an organization

P34 4.5.2 Add members

P34 4.5.3 Remove members

P34 4.6 Compression

P35 4.6.1 When is compression skipped?

P35 4.6.2 What happens to my Etag header?

P35 4.7 Regions

P35 4.8 Pricing & Limits

P36 4.9 Fair Use Policy

P36 4.9.1 Examples of Fair Use

P36 4.9.2 Not Fair Use

P36 4.9.3 Guidelines

P36 4.10 Privacy Policy

P36 4.10.1 What data do we collect?

P37 4.10.2 How do we collect your data?

P37 4.10.3 How will we use your data?

P37 4.10.4 How do we store your data?

P37 4.10.5 Marketing

P37 4.10.6 What are your data protection rights?

P37 4.10.7 Cookies

P37 4.10.8 How do we use cookies?

P37 4.10.9 What types of cookies do we use?

P38 4.10.10 How to manage cookies

P38 4.10.11 Privacy policies of other websites

P38 4.10.12 Changes to our privacy policy

P38 4.10.13 How to contact us

P38 4.11 Security & Responsible Disclosure

P38 4.11.1 Reporting a vulnerability

P38 4.11.2 Please do the following:

P38 4.11.3 Our commitment

P39 5 Runtime

P39 5.1 API Reference

P39 5.1.1 Web APIs

P39 5.1.2 Deno APIs

P40 5.1.3 Future support

P40 5.1.4 Limitations

P40 5.2 BroadcastChannel

P40 5.2.1 Constructor

P40 5.2.2 Properties

P40 5.2.3 Methods

P41 5.2.4 Example

P41 5.3 Fetch API

P42 5.3.1 fetch()

P42 5.3.2 Examples

P42 5.4 Request

P43 5.4.1 Constructor

P43 5.4.2 Properties

P43 5.4.3 Methods

P44 5.4.4 Example

P44 5.5 Response

P44 5.5.1 Constructor

P45 5.5.2 Properties

P45 5.5.3 Methods

P45 5.5.4 Example

P45 5.6 Headers

P45 5.6.1 Constructor

P46 5.6.2 Methods

P46 5.6.3 Example

P46 5.7 Sockets API

P46 5.7.1 Deno.connect

P47 5.7.2 Deno.connectTls

P47 5.8 File System API

P48 5.8.1 Deno.cwd

P48 5.8.2 Deno.readDir

P48 5.8.3 Deno.readFile

P49 5.8.4 Deno.readTextFile

P49 5.8.5 Deno.open

P50 5.8.6 Deno.File

P50 5.8.7 Deno.stat

P50 5.8.8 Deno.lstat

P51 5.8.9 Deno.FileInfo

P51 5.8.10 Deno.realPath

P51 6 Additional Resources

P51 6.1 HTTP Frameworks

P51 6.1.1 oak

P51 6.1.2 router

P51 6.1.3 nanossr

P51 6.1.4 sift

P52 6.1.5 fresh