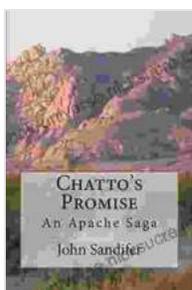


Chatto Promise: An Apache Saga

Chatto Promise is an Apache Saga, a distributed transaction framework for Apache Flink. It provides a simple and expressive API for building reliable and scalable distributed applications.

A saga is a long-running transaction that consists of a series of steps. Each step can be executed independently, but if any step fails, the entire saga must be rolled back.

Chatto Promise makes it easy to build sagas by providing a set of primitives that can be used to create and manage sagas. These primitives include:



Chatto's Promise: An Apache Saga by John Sandifer

★★★★☆ 4.4 out of 5

Language : English
File size : 21947 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Word Wise : Enabled
Print length : 300 pages
Lending : Enabled



- **SagaManager** : The SagaManager is responsible for creating and managing sagas.
- **SagaStep** : A SagaStep is a single step in a saga.

- **SagaContext** : The SagaContext provides access to the current saga and its state.

Chatto Promise offers a number of features that make it a valuable tool for building distributed applications, including:

- **Simplicity**: Chatto Promise provides a simple and expressive API for building sagas. This makes it easy to get started with Chatto Promise and to build complex sagas.
- **Reliability**: Chatto Promise ensures that sagas are executed reliably. If any step in a saga fails, the entire saga will be rolled back.
- **Scalability**: Chatto Promise is designed to be scalable. It can be used to build sagas that are executed across multiple machines.
- **Extensibility**: Chatto Promise is extensible. It can be used to build custom saga steps and to integrate with other systems.

To use Chatto Promise, you first need to create a **SagaManager** . The **SagaManager** is responsible for creating and managing sagas.

Once you have created a **SagaManager** , you can create a saga by calling the **createSaga()** method. The **createSaga()** method takes a **SagaDefinition** as an argument. The **SagaDefinition** defines the steps in the saga and the compensation actions that should be executed if any step fails.

Once you have created a saga, you can execute it by calling the **execute()** method. The **execute()** method takes a **SagaContext** as

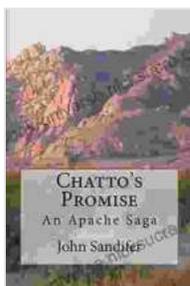
an argument. The **SagaContext** provides access to the current saga and its state.

The **SagaContext** can be used to execute the steps in the saga. Each step in the saga is represented by a **SagaStep**. The **SagaStep** provides a method called **execute()** that can be used to execute the step.

If any step in the saga fails, the **SagaManager** will automatically roll back the entire saga. The compensation actions that are defined in the **SagaDefinition** will be executed to undo the effects of the failed steps.

Chatto Promise is a powerful tool for building reliable and scalable distributed applications. It provides a simple and expressive API for creating and managing sagas. Chatto Promise is also extensible, so it can be used to build custom saga steps and to integrate with other systems.

If you are looking for a way to build reliable and scalable distributed applications, I encourage you to give Chatto Promise a try.

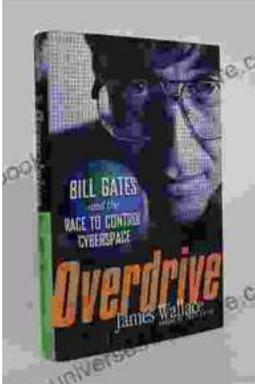


Chatto's Promise: An Apache Saga by John Sandifer

★★★★☆ 4.4 out of 5

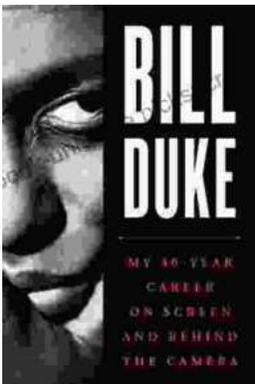
Language	: English
File size	: 21947 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Word Wise	: Enabled
Print length	: 300 pages
Lending	: Enabled





The Race to Control Cyberspace: Bill Gates's Plan for a Digital Divide

Bill Gates has a vision for the future of the internet. In his book, *The Road Ahead*, he argues that the internet will become increasingly important...



My 40 Year Career On Screen And Behind The Camera

I've been working in the entertainment industry for over 40 years, and in that time I've had the opportunity to work on both sides of the camera. I've...