



イベント処理をスマート・コントラクトを使って実装

土屋 健

表1 ファンクション・テーブル ds_function

項目	内容
id	登録処理を識別する ID 文字列
name	表示名称
url	パス情報。リクエスト・パスがこのパス文字列と前方一致した場合にイベント起動候補となる。 "*" 指定は全てのパスに一致する
method	HTTPリクエスト・メソッド。メソッドが一致した場合にイベント起動候補となる。 "*" 指定は全てのメソッドに一致する
act_pre	リクエスト処理実行前にイベント処理を実施するかどうか
act_post	リクエスト処理実行後にイベント処理を実施するかどうか
func_id	起動する処理を func_id で指定。 ds_function テーブルと対応

● イベント処理を作る理由

データ・サーバへのアクセスを契機に、定義済みの処理を起動するイベント機能を組み込みます。処理の実行環境は、

- サーバが動作している環境のローカル・コマンド
- 別のウェブ・アプリを呼び出す

など幾つかの方式が考えられますが、データ・サーバではスマート・コントラクトをイベント処理に使います。

イベント処理をスマート・コントラクトを使って実現する理由は以下です。

- ブロックチェーン技術の上に成り立っており安定した実行環境である
- 実行記録が残りに不正が生じない
- イベント処理で記録するデータがブロックチェーンに入るため改ざんの恐れがない

実際のシステムでは、適材適所でさまざまなものを組み合わせるので、全てのイベントをスマート・コントラクトで処理することはありません。

こんなイベント処理を作る

● 仕様

イベントの仕様を決めます。

- データ・サーバへのリクエストを契機に登録処理が実行される仕組みとする
- イベント処理対象のリクエストは、URL+HTTPリクエスト・メソッドで指定する
- リクエスト処理の前後でイベント処理を行うタイミングがある
- 処理はEthereum上でスマート・コントラクトとして実装&実行
- イベント処理として登録できるコントラクトは、put、getの関数を持つものとする(リスト1)。

● 制御

以下の2つのテーブルでイベント起動の設定を行います。

- イベントの発生条件と起動する処理を定義するイベント・テーブル(ds_event)
- 起動する処理を定義するファンクション・テーブル(ds_function)

ファンクション・テーブル(ds_function)については、第4章で説明したので、ここではイベント・テーブル(ds_event)について説明します。

● イベント・テーブル(ds_event)

コントラクトを実行するためには、ABIとコントラクトのアドレス情報が必要です。これらの情報はコントラクト・プログラムを修正し、デプロイするたびに変わるものなので、プログラムの中に埋め込むことはできません。そのため、それらの情報をパラメータ化してコントラクトを呼び出すプログラムで参照します。そのパラメータはMySQLに保存し、ファンクション・テーブル(ds_function)で管理します。ファンクション・テーブル(ds_function)は、表1の項目を持ちます。