



# AWS SDK for PHP 概要

## AWS Black Belt Online Seminar

Tomohiro Kamitani

Solutions Architect

2022/06

# AWS Black Belt Online Seminarとは

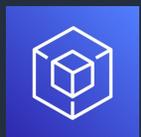
- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWSの技術担当者が、AWSの各サービスやソリューションについてテーマごとに動画を公開します
- 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます
- 以下のURLより、過去のセミナー含めた資料などをダウンロードすることができます
- <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>

# 内容についての注意点

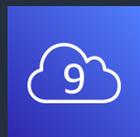
- 本資料では2022年06月時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<https://aws.amazon.com/>)にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます

# 自己紹介

- 名前  
紙谷知弘 (かみたに ともひろ)
- 所属  
アマゾンウェブサービスジャパン株式会社  
技術統括本部  
エンタープライズ技術本部  
メディアソリューション部  
ソリューションアーキテクト
- 好きなAWSサービス



AWS SDK



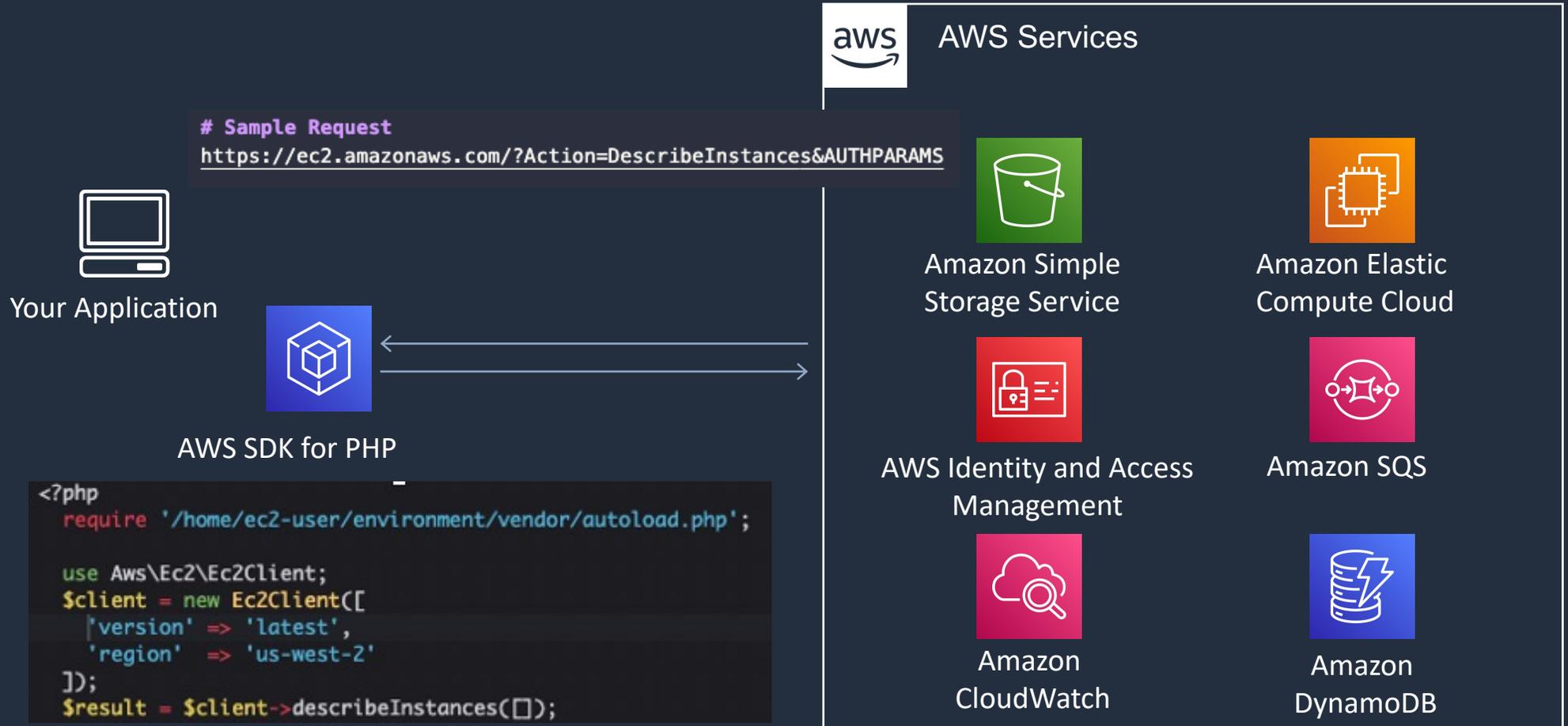
AWS Cloud9



# 本セッションでお話しすること

1. AWS SDK for PHP のインストール方法
2. 関連ドキュメントのご紹介
3. 基本的な使い方 / サービス呼び出しの流れ

# AWS SDK for PHPとは？



# インストール方法

- Composerを利用
  1. Composerをインストール
  2. `composer require aws/aws-sdk-php`を実行
  3. `autoload.php`をインクルード
- パッケージ済みのphar(PHPアーカイブ)を利用
  1. pharをダウンロードしてインクルード
- ZIPファイルを利用
  1. ZIPファイルをダウンロードしてaws-autoloader.phpをインクルード

[https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/getting-started\\_installation.html](https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/getting-started_installation.html)

# 互換性テスト

## CLI版

```
=====  
AWS SDK for PHP Compatibility Test  
=====
```

System requirements

- Ensuring that the version of PHP is >= 5.5.0: [OK]
- Ensuring that detect\_unicode is disabled: [OK]
- Ensuring that session.auto\_start is disabled: [OK]
- Checking if the pcre extension is installed: [OK]
- Checking if the spl extension is installed: [OK]
- Checking if the json extension is installed: [OK]
- Checking if the simplexml extension is installed: [OK]
- Ensuring that cURL can send https requests: [OK]
- Ensuring that file\_get\_contents works: [OK]

System recommendations

- Checking if you are running on a 64-bit platform: [OK]
- Ensuring that zend.enable\_gc is enabled: [OK]
- Ensuring that date.timezone is set: [WARNING]
  - The date.timezone PHP ini setting has not been set in /etc/php.ini
- Checking if the dom extension is installed: [OK]
- Checking if the curl extension is installed: [OK]
- Checking if the openssl extension is installed: [OK]
- Checking if the zlib extension is installed: [OK]
- Checking if OPcache is enabled: [WARNING]
  - opcache.enable in /etc/php.ini is currently set to false but should be set to 1.

## Web版

### AWS SDK for PHP Compatibility Test

#### System requirements

- Ensuring that the version of PHP is >= 5.5.0
- Ensuring that detect\_unicode is disabled
- Ensuring that session.auto\_start is disabled
- Checking if the pcre extension is installed
- Checking if the spl extension is installed
- Checking if the json extension is installed
- Checking if the simplexml extension is installed
- Ensuring that cURL can send https requests
- Ensuring that file\_get\_contents works

#### System recommendations

- Checking if you are running on a 64-bit platform
- Ensuring that zend.enable\_gc is enabled

Ensuring that date.timezone is set: [WARNING]

The date.timezone PHP ini setting has not been set in

SDKに付属しているCompatibility-test.phpを実行して、対象の環境でSDKを実行することができるかどうかを確認

# フレームワークとの連携

- Laravelとの連携
  - AWSサービスプロバイダーが提供されている
  - Composerでインストールした後サービスプロバイダに登録することにより利用可能
  - 詳細は<https://github.com/aws/aws-sdk-php-laravel>を確認
- Symfonyとの連携
  - AWSバンドルが提供されている
  - Composerでインストールした後バンドルを有効化することにより利用可能
  - 詳細は<https://github.com/aws/aws-sdk-php-symfony>を確認

# AWS SDK for PHP の設計について

- AWS SDK for PHPの設計はCommand patternであり動作がカプセル化されている
- クライアントの中身を見てみるとわかるがメソッドの実態は存在しないことがわかる

暗黙的なコマンド呼び出し

```
$result = $s3Client->putObject($params);
```

明示的なコマンド呼び出し

```
$command = $s3Client->getCommand('PutObject', $params);  
$result = $s3Client->execute($command);
```

どちらも同じ結果

[https://docs.aws.amazon.com/ja\\_jp/sdk-for-php/v3/developer-guide/guide\\_commands.html](https://docs.aws.amazon.com/ja_jp/sdk-for-php/v3/developer-guide/guide_commands.html)

# 開発を始める前に(ドキュメントの歩き方その1)

## 対象サービスのデベロッパーガイド

対象のサービスの説明と共にサンプルコードが載っているのでそちらを参考にしながら実装できる(サービスによっては特定の言語のみの場合もあり。リンクはs3の例)

[https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/userguide/UsingTheMPphpAPI.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/UsingTheMPphpAPI.html)

## AWS SDK for PHPAPI Documentation

各サービスのクライアントのメソッドの呼び出し方の詳細を確認するならこちらを参照。  
必須パラメーターを事前に確認しておくスムーズ

<https://docs.aws.amazon.com/aws-sdk-php/v3/api/>

# 開発を始める前に(ドキュメントの歩き方その2)

## GitHubリポジトリ

SDK特有の機能の実装を確認したいのであればこちらを参照。  
ドキュメントの説明だけでは足りない部分をPHP docで補完できる  
<https://github.com/aws/aws-sdk-php>

## GitHubサンプルコード リポジトリ

こちらに各サービスの呼び出しのサンプルコードがまとまっている。  
呼び出し方で迷ったらこちらを参照。  
[https://github.com/awsdocs/aws-doc-sdk-examples/tree/master/php/example\\_code](https://github.com/awsdocs/aws-doc-sdk-examples/tree/master/php/example_code)

## AWS re:Post

インストールでつまづいた。  
コードがエラーを吐いてうまくいかないなどお困りごとが発生したらこちら。  
<https://repost.aws/>

# AWS SDK for PHP API Documentation

- トップページにサービス名とクラス一覧がありリンクをクリックして詳細を確認できる
- サービスのトップページに操作メソッド一覧がありクリックすると詳細に飛ぶ
- メソッド詳細には必須パラメーターやレスポンスの内容が記載されている

The screenshot shows the AWS SDK for PHP API documentation for the `DescribeInstances` method. It features a table of service APIs on the left and a detailed view of the `DescribeInstances` method on the right.

Service Name	Client Class
<a href="#">Access Analyzer</a>	<a href="#">Aws\AccessAnalyzer\AccessAnalyzerClient</a>
<a href="#">AWS Certificate Manager</a>	<a href="#">Aws\Acm\AcmClient</a>
<a href="#">AWS Certificate Manager Private Certificate Authority</a>	<a href="#">Aws\ACMPCA\ACMPCAClient</a>
<a href="#">Alexa For Business</a>	<a href="#">Aws\AlexaForBusiness\AlexaForBusinessClient</a>
<a href="#">AWS Amplify</a>	<a href="#">Aws\Amplify\AmplifyClient</a>
<a href="#">AmplifyBackend</a>	<a href="#">Aws\Amplify\AmplifyBackend</a>
<a href="#">Amazon API Gateway</a>	<a href="#">Aws\ApiGateway\ApiGatewayClient</a>
<a href="#">AmazonApiGatewayManagementApi</a>	<a href="#">Aws\ApiGatewayManagementApi\ApiGatewayManagementApiClient</a>
<a href="#">AmazonApiGatewayV2</a>	<a href="#">Aws\ApiGatewayV2\ApiGatewayV2Client</a>
<a href="#">Amazon AppConfig</a>	<a href="#">Aws\AppConfig\AppConfigClient</a>
<a href="#">Amazon Appflow</a>	<a href="#">Aws\Appflow\AppflowClient</a>
<a href="#">Amazon AppIntegrations Service</a>	<a href="#">Aws\AppIntegrationsService\AppIntegrationsServiceClient</a>
<a href="#">Application Auto Scaling</a>	<a href="#">Aws\AutoScaling\AutoScalingClient</a>
<a href="#">AWS Application Cost Profiler</a>	<a href="#">Aws\CostProfiler\CostProfilerClient</a>
<a href="#">AWS Application Discovery Service</a>	<a href="#">Aws\Discovery\DiscoveryClient</a>
<a href="#">Amazon CloudWatch Application Insights</a>	<a href="#">Aws\CloudWatch\CloudWatchClient</a>

**DescribeInstances**

```
$result = $client->describeInstances(['* ... *']);
$promise = $client->describeInstancesAsync(['* ... *']);
```

Describes the specified instances or all instances.

If you specify instance IDs, the output includes information for only the specified instances. If you do not specify instance IDs, the output includes information for all instances, which can affect performance. We recommend that you use the `Filters` parameter to filter the results.

If you specify an instance ID that is not valid, an error is returned. If you specify an instance that you have recently terminated, the instance might appear in the returned results. This interval is usually less than 15 minutes.

If you describe instances in the rare case where an Availability Zone is experiencing a service disruption, only instance IDs that are in an unaffected zone, the call works normally.

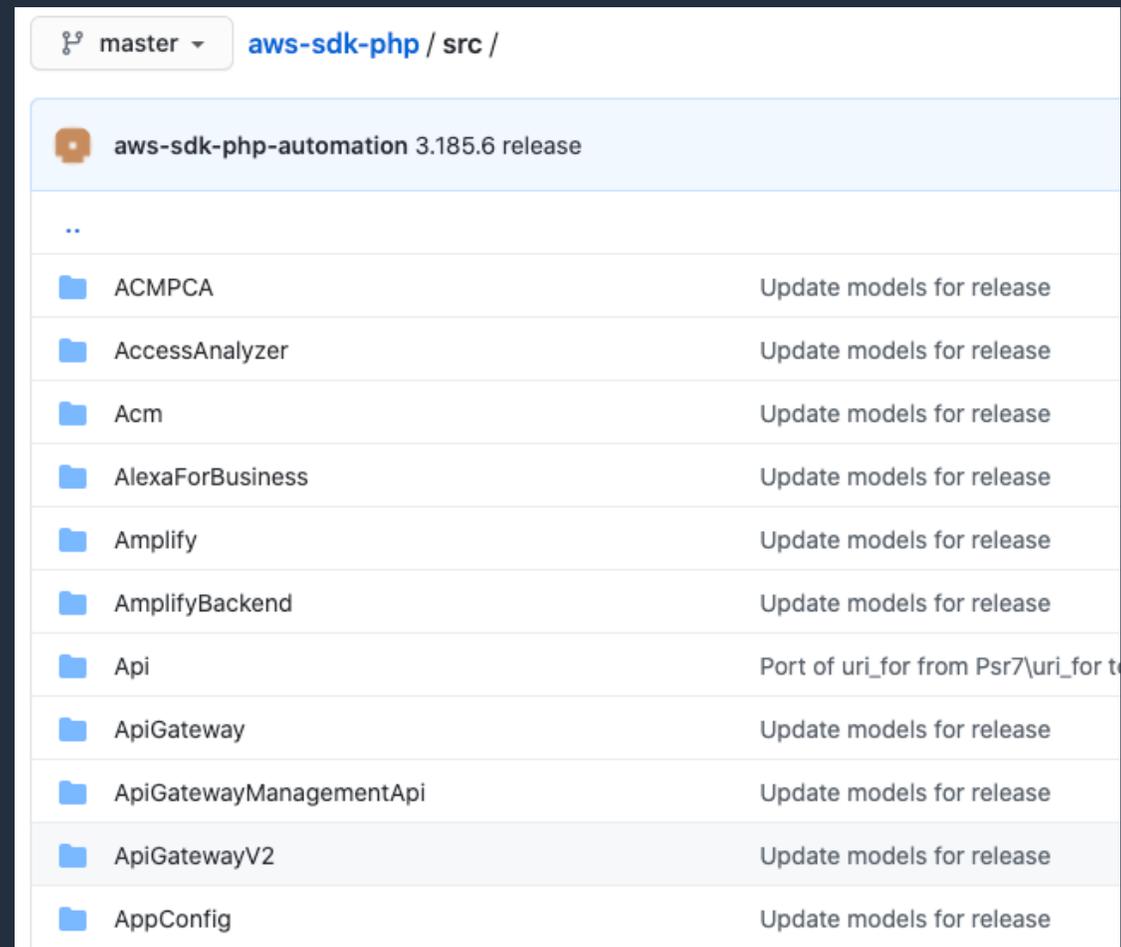
**Parameter Syntax**

```
$result = $client->describeInstances([
    'DryRun' => true || false,
    'Filters' => [
        [
            'Name' => '<string>',
            'Values' => ['<string>', ...],
        ],
    ],
]);
```

<https://docs.aws.amazon.com/aws-sdk-php/v3/api/>

# GitHub リポジトリ

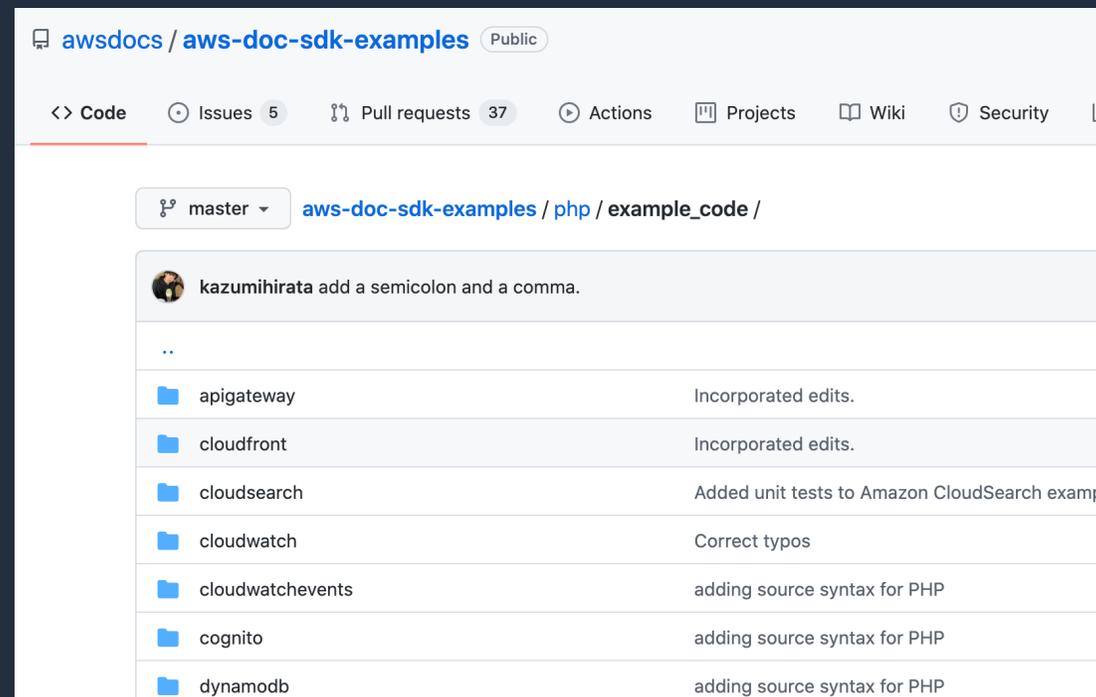
- src 配下に各サービスのクラスファイルが配置されている。また基本クラスのファイルも同階層に配置されているため aws sdk for php 特有の機能について詳しく知りたければそちらを参照
- バグ/機能改修についてはissueを受けてつけており、Pull Requestで貢献していただくこともできます



<https://github.com/aws/aws-sdk-php>

# GitHub サンプルコードリポジトリ

- どのようにAWS SDK for PHPをつかって色々なサービス呼び出すのかが紹介されている
- 一部サービスの各機能呼び出しのサンプルコードが提供されている
- いくつかのコードではユニットテストも付属しており参考にすることができる



[https://github.com/awsdocs/aws-doc-sdk-examples/tree/master/php/example\\_code](https://github.com/awsdocs/aws-doc-sdk-examples/tree/master/php/example_code)

# コード例

例) EC2 インスタンスの一覧の取得

1. AWSサービスに応じたクライアントを作成  
コンストラクタの引数でデバッグの有効化やリトライ回数などを指定できる。

```
require 'vendor/autoload.php';  
  
use Aws\Ec2\Ec2Client;
```

```
$ec2Client = new Aws\Ec2\Ec2Client([  
    'region' => 'us-west-2',  
    'version' => '2016-11-15',  
    'profile' => 'default'  
]);
```

2. メソッド呼び出し  
引数を指定してメソッドを呼び出す。引数には必須のものもあるのでAPI docをみて確認

```
$result = $ec2Client->describeInstances();  
  
var_dump($result);
```

<https://docs.aws.amazon.com/sdk-for-php/v3/developer-guide/ec2-examples-managing-instances.html>

# Sdkクラスの利用

- 各サービスのクライアントのコンストラクタにオプションの連想配列を渡すことによって、クライアントを作成できます
- Sdkクラスはクライアントファクトリとして機能し複数のクライアント間で共有される設定オプションを管理できます
- アプリケーション内で複数のクライアントインスタンスを利用する場合はSdkクラスを利用することを推奨しています

## クライアントの作成

```
require 'vendor/autoload.php';
use Aws\S3\S3Client;
use Aws\Exception\AwsException;

$s3 = new Aws\S3\S3Client([
    'version' => 'latest',
    'region' => 'us-east-2'
]);
```

## Sdkクラスを用いた例

```
require 'vendor/autoload.php';
use Aws\S3\S3Client;
use Aws\Exception\AwsException;

$sharedConfig = [
    'region' => 'us-west-2',
    'version' => 'latest'
];

$sdk = new Aws\Sdk($sharedConfig);

$client = $sdk->createS3();
```

[https://docs.aws.amazon.com/ja\\_jp/sdk-for-php/v3/developer-guide/getting-started\\_basic-usage.html](https://docs.aws.amazon.com/ja_jp/sdk-for-php/v3/developer-guide/getting-started_basic-usage.html)



# サービスオペレーションの実行

- サービスオペレーションは、クライアントオブジェクトにあるメソッドを呼び出すことによって実行できます
- Asyncが付いているメソッドを呼び出すとリクエストを非同期的に呼び出すことができます

例) 同期リクエスト

```
$result = $s3Client->putObject([
    'Bucket' => 'my-bucket',
    'Key'    => 'my-key',
    'Body'   => 'this is the body!',
    '@http' => [
        'proxy' => 'http://192.168.16.1:10'
    ]
]);
```

例) 非同期リクエスト

```
$promise = $s3Client->listBucketsAsync();
$promise
    ->then(function ($result) {
        echo 'Got a result: ' . var_export($result, true);
    })
    ->otherwise(function ($reason) {
        echo 'Encountered an error: ' . $reason->getMessage();
    });
```

# コンストラクタのオプション

設定オプション	概要
debug	trueに設定するとデバッグ情報を出力します。デバッグ情報にはトランザクションの状態の変更に関する情報が含まれています。
stats	転送の統計情報をオペレーションによって返される結果とエラーにバインドします。trueに設定すると送信されたリクエストに関する統計情報が収集されます。
retries	クライアントの再試行を設定します。0を設定すると再試行が無効になります。
profile	認証情報が作成されるときに使用するプロファイルを指定できます。この設定は環境変数のAWS_PROFILEをオーバーライドします。
endpoint	MediaConvertなどのアカウント固有のエンドポイントを使用するサービスに必要です。

※上記の表は下記リンクより一部抜粋

[https://docs.aws.amazon.com/ja\\_jp/sdk-for-php/v3/developer-guide/guide\\_configuration.html](https://docs.aws.amazon.com/ja_jp/sdk-for-php/v3/developer-guide/guide_configuration.html)

# エラー処理 (同期エラー)

- 同期エラー処理の場合コード内でエラーを処理する必要がある場合は、オペレーションをtry/catchブロック内に配置します。
- エラーが発生するとサービス固有の例外がスローされます。

## エラー処理の例

```
$sdk = new Aws\Sdk([
    'region' => 'us-west-2',
    'version' => 'latest'
]);

$s3Client = $sdk->createS3();

try {
    $s3Client->createBucket(['Bucket' => 'my-bucket']);
} catch (S3Exception $e) {
    echo $e->getMessage();
} catch (AwsException $e) {
    echo $e->getAwsRequestId() . "\n";
    echo $e->getAwsErrorType() . "\n";
    echo $e->getAwsErrorCode() . "\n";

    var_dump($e->toArray());
}
```

# エラー処理 (非同期エラー)

- 非同期リクエスト(Asyncがついているメソッド)の場合は例外はスローされません。
- 代わりに返されたpromiseのthen()またはotherwise()メソッドを使用して、結果またはエラーを受け取る必要があります。

## 非同期エラー処理の例

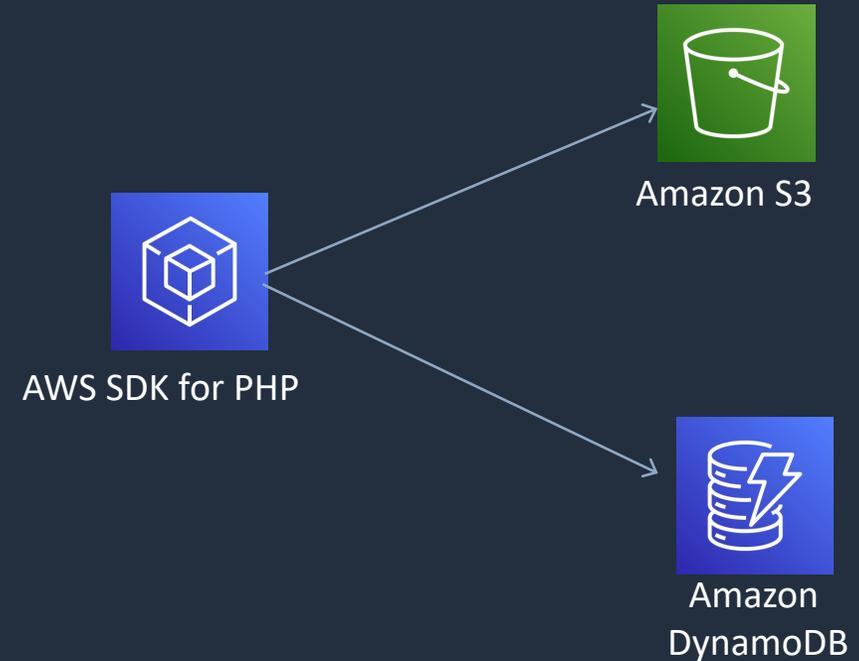
```
$promise = $s3Client->createBucketAsync(['Bucket' => 'my-bucket']);  
$promise->otherwise(function ($reason) {  
    var_dump($reason);  
});  
  
$promise->then(null, function ($reason) {  
    var_dump($reason);  
});
```

## ラップ解除の例

```
try {  
    $result = $promise->wait();  
} catch (S3Exception $e) {  
    echo $e->getMessage();  
}
```

# 各サービスとの連携

- サービスによっては高レベルAPIやオペレーションを実行する以外の追加機能が提供されています
- DynamoDB
  - セッションハンドラー
- Amazon S3
  - マルチリージョンクライアント
  - ストリームラッパー
  - Transfer Manager



[https://docs.aws.amazon.com/ja\\_jp/sdk-for-php/v3/developer-guide/service\\_index.html](https://docs.aws.amazon.com/ja_jp/sdk-for-php/v3/developer-guide/service_index.html)

# 高レベルAPIとは？

高レベルAPIとは処理がより抽象化され一般的なユースケースを簡素化したもの

低レベルAPIでの  
S3 マルチパートアップロー  
ド

```
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket' => $bucket,
    'Key' => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata' => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket' => $bucket,
            'Key' => $keyname,
            'UploadId' => $uploadId,
            'PartNumber' => $partNumber,
            'Body' => fread($file, 5 * 1024 * 1024),
        ]);
        $parts['Parts'][$partNumber] = [
            'PartNumber' => $partNumber,
            'ETag' => $result['ETag'],
        ];
        $partNumber++;
        echo "Uploading part {$partNumber} of {$filename}." . PHP_EOL;
    }
    fclose($file);
}
```

続く . . .



高レベルAPIでの  
S3 マルチパートアップロー  
ド

```
require 'vendor/autoload.php';

use Aws\Common\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key' => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

[https://docs.aws.amazon.com/ja\\_jp/AmazonS3/latest/userguide/mpuoverview.html](https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/mpuoverview.html)

# 本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料のAWSサポート窓口へお問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください  
#awsblackbelt

# その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!