

Расчет контрольной суммы

Для некоторых методов требуется указать дополнительный параметр безопасности sig. В интерфейсе вы сможете проверить алгоритм расчета этого параметра на сервере. Разработчику приложения нужно реализовать свой собственный метод для получения данного параметра.

The screenshot shows a web browser window with the URL `api.tpu.ru/dashboard/prilogenie/platforma/version/signature.html?parent_id=81&id=101#`. The page title is "Расчет подписи параметров" (Signature Calculation). It features a form with two columns: "Название параметра *" (Parameter Name) and "Значение *" (Value). The first row contains "id" and "123". The second row contains "ip" and "123.100.10.15". There are red "X" buttons next to each value field. Below the form is a blue button "+ Добавить параметр" (Add parameter) and a green button "✓ Рассчитать контрольную сумму" (Calculate checksum). The result section, titled "Результат подсчета" (Calculation result), shows a table with a single row: "sig" with the value "9e5a6f4459cb0ec0d9f8643cf920811f".

Название параметра *	Значение *
id	123
ip	123.100.10.15

+ Добавить параметр

✓ Рассчитать контрольную сумму

Результат подсчета	
sig	9e5a6f4459cb0ec0d9f8643cf920811f

Пример на языке [PHP](#)

```
public function sig(array $arguments, $secret)
{
    ksort($arguments);
    $string = '';

    foreach ($arguments as $key => $value) {
        $string .= "$key=$value";
    }

    return md5($string . $secret);
}
```

Пример на языке [Java](#)

```
import java.security.MessageDigest;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;
import java.security.NoSuchAlgorithmException;

public String sig(Map<String, String> arguments, String secret) throws NoSuchAlgorithmException {
    StringBuilder builder = new StringBuilder();

    List<String> keys = arguments.keySet().stream().sorted().collect(Collectors.toList());

    for(String key : keys) {
        builder.append(key).append("=").append(arguments.get(key));
    }

    builder.append(secret);
    byte[] digest = MessageDigest.getInstance("MD5").digest(builder.toString().getBytes());

    StringBuilder resultBuilder = new StringBuilder();

    for(byte b : digest) {
        resultBuilder.append(String.format("%02x", b));
    }

    return resultBuilder.toString();
}
```

Пример на языке [Scala](#)

```
import java.security.MessageDigest
import scala.collection.immutable.ListMap

def sig(arguments: Map[String, String], secret: String): String = {
    MessageDigest.getInstance("MD5").digest {
        s"${ListMap(arguments.toSeq.sortBy({ case (key, value) => key }):_*).map({ case (key, value) =>
s"$key=$value" }).mkString}$secret".getBytes
    }.map(byte => String.format("%02x", byte)).mkString
    }
}
```

Пример на языке [Python](#)

```
import collections
import hashlib

def sig(arguments, secret):
    ordered = collections.OrderedDict(sorted(arguments.items()))
    concated = "".join(list(map(lambda k,v: k + "=" + v, ordered.keys(), ordered.values())) + secret)
    return hashlib.md5(concated.encode()).hexdigest()
```

Revision #4

Created 3 October 2024 08:06:40 by Дмитрий Евгеньевич Семенов

Updated 3 October 2024 08:14:21 by Дмитрий Евгеньевич Семенов