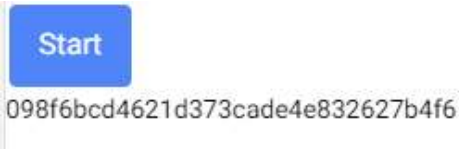


Client側(Typescript)

```
npm install ts-md5 --save
```

```
import {Md5} from 'ts-md5/dist/md5';
```

```
this.data = Md5.hashStr("test").toString();
```



・Saltと複数回で暗号化を組み合わせ
使ってください。
・サーバ側に処理がないですが、必要
があれば、復号化しにくい処理を追加
してください。

DB側(postgres)

```
4 select md5('test');
```

数据输出 解释 消息 Notifications Query History

md5
text
1 098f6bcd4621d373cade4e832627b4f6

UUID定数を生成

```
SELECT uuid_generate_v3(uuid_ns_oid(), '098f6bcd4621d373cade4e832627b4f6');  
⇒ 5f63ac5e-52e0-320f-b5d2-da4f06b10ba9
```

参照:

uuid-oss: <https://www.postgresql.jp/document/9.4/html/uuid-oss.html>
ts-md5: <https://www.npmjs.com/package/ts-md5>

base64について、

Typescript

```
npm install --save js-base64
```

```
import { Base64 } from 'js-base64';
```

```
Base64.encode('abc')
```

参照: <https://www.npmjs.com/package/js-base64>

Java

```
import java.util.Base64;
```

```
System.out.println(Base64.getEncoder().encodeToString("tested".getBytes()));  
System.out.println(new String(Base64.getDecoder().decode("dGVzdGVk")));
```

参照: <https://gist.github.com/komiya-atsushi/d878e6e4bf9ba6dae8fa>

Postgres

```
encode(data bytea, type text)  
decode(string text, type text)
```

参照: <https://www.postgresql.jp/document/7.2/user/functions-string.html>