

# Security Questions Module - tKey JS SDK

[@tkey/security-questions](#)

[â](#)

The Security Questions Module helps you add or remove the and password as a share for tkey.

## Installationâ

- npm
- Yarn
- pnpm

npm install --save @tkey/security-questions yarn add @tkey/security-questions pnpm add @tkey/security-questions

## Initializationâ

### Parametersâ

params

- saveAnswers?
- :boolean

import

{

SecurityQuestionsModule

}

from

"@tkey/security-questions" ; const securityQuestionsModule =

new

SecurityQuestionsModule ( ) ;

## Usageâ

With theSecurityQuestionsModule , you have access to the multiple functions as mentioned in the type reference, however, the most important ones are:

### Generate New Share with Question and Answerâ

generateNewShareWithSecurityQuestions(answerString: string, questions: string): Promise;

- answerString
- : Answer corresponding to a security question
- questions
- : The security question,must be greater than 10 characters

### Exampleâ

await ( tKeyInstance . modules . securityQuestions

as

SecurityQuestionsModule ) . generateNewShareWithSecurityQuestions ( password ,

"whats your password?" ) ;

### Input Existing Security Questions Share into tKey Instanceâ

inputShareFromSecurityQuestions(answerString: string): Promise;

### Example[^](#)

- answerString
- : Answer corresponding to a security question

await ( tKeyInstance . modules . securityQuestions

as

any ) . inputShareFromSecurityQuestions ( password ) ;

### Change Security Question and Answer[^](#)

changeSecurityQuestionAndAnswer(newAnswerString: string, newQuestions: string): Promise;

- newAnswerString
- : Answer corresponding to the new security question
- newQuestions
- : The new security question

### Example[^](#)

await ( tKeyInstance . modules . securityQuestions

as

any ) . changeSecurityQuestionAndAnswer ( password ,

"whats your password?" ) ;

## Type Reference[^](#)

### SecurityQuestionsModule

[^](#)

class

SecurityQuestionsModule

implements

IModule

{ moduleName :

string ; tbSDK :

ITKeyApi ; saveAnswers :

boolean ; constructor ( saveAnswers ? :

boolean ) ; static

refreshSecurityQuestionsMiddleware ( generalStore :

unknown , oldShareStores :

ShareStoreMap , newShareStores :

ShareStoreMap ) :

unknown ; setModuleReferences ( tbSDK :

ITKeyApi ) :

void ; initialize ( ) :

Promise < void

```

        ; generateNewShareWithSecurityQuestions ( answerString :
string , questions :
string ) :
Promise < GenerateNewShareResult
        ; getSecurityQuestions ( ) :
string ; inputShareFromSecurityQuestions ( answerString :
string ) :
Promise < void
        ; changeSecurityQuestionAndAnswer ( newAnswerString :
string , newQuestions :
string ) :
Promise < void
        ; saveAnswerOnTkeyStore ( answerString :
string ) :
Promise < void
        ; getAnswer ( ) :
Promise < string
        ; }

```

## GenerateNewShareResult

[â](#)

```

declare
type
GenerateNewShareResult
=
{ newShareStores :
ShareStoreMap ; newShareIndex :
BN ; } ; export
declare
type
ShareStoreMap
=
{ [ shareIndex :
string ] :
ShareStore ; } ; declare
class
ShareStore
implements

```

ISerializable

{ share :

Share ; polynomialID :

PolynomialID ; constructor ( share :

Share , polynomialID :

PolynomialID ) ; static

fromJSON ( value :

StringifiedType ) :

ShareStore ; toJSON ( ) :

StringifiedType ; } [Edit this page](#) [Previous](#) [Chrome Storage](#) [Next](#) [Share Transfer](#)