

# Вход через социальные сети






Бесплатный вход через социальные сети с простой интеграцией

- [Процесс авторизации](#)
- [Решение для Web приложения или сайта](#)
- [Авторизация через социальные сети в CapacitorJS](#)
- [Авторизация через социальные сети в ReactNative](#)

# Процесс авторизации

Для авторизации через социальные сети вы можете использовать **БЕСПЛАТНОЕ** решение от Auth4App, интеграция максимально упрощает использование социальных сетей в вашем приложении или на сайте.

## Поддерживаемые соц. сети

		
yandex	vk	google
		Дополнительные соц. сети может добавить по запросу.
ok	mailru	

## Процесс авторизации

Для авторизации через соц. сети следует вызвать ссылку вида

```
https://api.auth4app.com/auth?soc={КЛЮЧ СОЦ. СЕТИ}&key={ВАШ УНИКАЛЬНЫЙ КЛЮЧ}
```

Где soc это ключ социальной сети из таблицы выше, а уникальный ключ (минимум 256 символов) это уникальный ключ, по которому можно будет получить результат авторизации.

После успешной авторизации произойдет переход на страницу:

<https://api.auth4app.com/callback/success>

Которая будет автоматически закрыта



Если авторизация была отменена или прошла с ошибкой то будет переход на страницу:

<https://api.auth4app.com/callback/error>

Которая будет автоматически закрыта



## Получение результат авторизации

```
https://api.auth4app.com/hash?key={ВАШ УНИКАЛЬНЫЙ КЛЮЧ}
```

Сделав запрос с ключом авторизации который вы создавали для ссылки авторизации вы получите результат авторизации. А сам ключ действителен 1 минуту после авторизации.

**Ответ в случае если авторизация прошла с ошибкой или не была пройдена:**

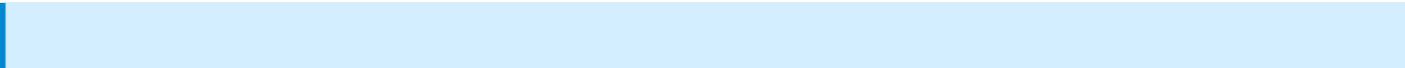
```
{
  "type": "error",
  "data": [ ]
}
```

**Успешная авторизация:**

```
{
  "type": "success",
  "data": {
    "id": "406873ad05c4285____79a95da33a9a124d3",
    "profile_id": "11254____6136911",
    "type": "google",
    "mail": "mail@gmail.com",
    "name": "Ivan",
    "surname": "Ivanov",
    "photo": "https://lh3.googleusercontent.com/a/ACg8____j=s96-c",
    "userInfo": {
      "id": "1125____36911",
      "email": "mail@gmail.com",
      "verified_email": true,
      "name": "Ivan Ivanov",
      "given_name": "Ivan",
      "family_name": "Ivanov",
      "picture": "https://lh3.googleusercontent.com/a/ACg8o____=s96-c",
      "locale": "ru"
    },
  }
}
```

Описание полей ответа:

Поле	Описание
id	Уникальный ID ключ пользователя, будет постоянным при авторизации пользователя, можно использовать как уникальный ключ пользователя при сохранении способа авторизации.
profile_id	ID из социальной сети в которой была авторизации
type	Ключ социальной сети, через которую была авторизация
mail	Email пользователя из социальной сети
name	Имя пользователя
surname	Фамилия пользователя
photo	Ссылка на фото профиля
userInfo	Оригинальный данные авторизации, которые были получены от социальной сети, отличаются в разных соц. сетях.



Используйте поля name, surname, mail и photo для заполнения профиля пользователя, они статичны для всех социальных сетей и автоматически подставляются из userInfo, где ключи данных отличаются.

# Решение для Web приложения или сайта

Для реализации в Web можно использовать открытие страницы браузера и делать обработку на момент ее закрытия.

## Алгоритм работы

1. Пользователь нажимаем войти через соц. сеть
2. Открывается окно с ссылкой на авторизацию
3. На событие закрытие окна авторизации (оно будет закрыто автоматически) делаем обработку с key для получения данных или запроса на сервер.

## Пример JavaScript кода

```
//Пример функции генерации уникального ключа

function getAlphaNumericRandom = (len){
  if (len == undefined || len <= 0) {
    len = 1;
  }
  var characters =
    'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
  var result = '';
  var iffirst = 0;
  for (var i = 0; i < len; i++) {
    if (i == 0) {
      iffirst = 10;
    } else {
      iffirst = 0;
    }
    result +=
      characters[Math.round(Math.random() * (characters.length - iffirst - 1))];
  }
}
```

```

    return result;
};

function generateUUID = (length) => {
    var baseStr = '';
    if (length) {
        for (var i = 0; i < length; i++) baseStr += 'x';
    } else {
        baseStr = 'xxxxxxxxxx';
    }
    var d = new Date().getTime();
    var uuid = baseStr.replace(/[xy]/g, function (c) {
        var r = (d + Math.random() * 16) % 16 | 0;
        d = Math.floor(d / 16);
        return (c == 'x' ? r : (r & 0x3 | 0x8)).toString(16);
    });
    return uuid;
};

//-----

let key = new Date().getTime() + '-' + generateUUID(256) + '-' + getAlphaNumericRandom(100);
let type = "vk";

let win = window.open(
    `https://auth4app.com/auth?soc=${type}&key=${key}`,
    '_blank'
);

const interval = setInterval(() => {
    if (win && win.closed) {
        clearInterval(interval);

        //Делаем запрос на сервер с key, на котором получаем данные через метод /hash
        //Или получаем данные локально через метод /hash

    }
}, 500);

```





# Авторизация через социальные сети в CapacitorJS

Для авторизации через социальные сети в гибридном мобильном приложении CapacitorJS, используйте официальный плагин: [capacitor-simple-social-auth4app](#)

Он создает нативное WebView для авторизации, сам генерирует ключ и возвращает его и данные пользователя в случае успешной авторизации.

```
SimpleSocialAuth4App.removeAllListeners();
SimpleSocialAuth4App.auth({ social: "vk" });
SimpleSocialAuth4App.addListener("authSuccess", (e) => {
  let key = e.key;
  //Логика работы с ключом для получения данных
  //Или запрос на сервер для авторизации
});

SimpleSocialAuth4App.addListener("authError", (e) => {
  //Ошибка авторизации
  console.log(e);
});
```

# Авторизация через социальные сети в ReactNative

Для реализации авторизации через социальные сети в ReactNative можно использовать компонент `WebView` для отображения формы входа пользователю и получение результата авторизации по ссылкам редиректа.

```
export const AuthScreen = () => {

  const key = "{УНИКАЛЬНЫЙ КЛЮЧ АВТОРИЗАЦИИ}";
  const soc = "vk"; //Ключ соц. сети

  const onNavigationStateChange = (e) => {

    if (e.url.indexOf('callback/error') > -1) {
      //Ошибка авторизации
    }
    if (e.url.indexOf('callback/success') > -1) {
      //Авторизация прошла успешно
      //Логика работы с key для получения данных локально
      //Или запрос на сервер
    }
  };

  const initUrl = `https://api.auth4app.com/auth?soc=${soc}&key=${key}`;

  return (
    <>
    <WebView
```

```
source={{  
  uri: initUrl,  
}}  
style={{flex: 1}}  
onNavigationStateChange={onNavigationStateChange}  
/>  
</>  
);  
}
```