



Фронтенд - Авторизация

Права можно применять к маршрутам и элементам разметки.

[Права на маршрут](#)

[Права на элемент разметки](#)

Права на маршрут

В файле `router/routes.js` для маршрута можно указать следующие *meta*-свойства:

- **authorizeRoles** - массив ролей, наличие одной из которых дает пользователю доступ к маршруту;

```
// Пример
{
  path: '/reports',
  name: 'reports',
  component: () => import('../views/Reports.vue'),
  // Страница отчетов доступна проверяющим и администраторам
  meta: { title: 'Отчёты', authorizeRoles: [Role.Supervisor, Role.AdminTN, Role.Admin] },
},
```

- **authorizeFuncs** - массив функций, наличие одной из которых дает пользователю доступ к маршруту.

Если есть у пользователя есть необходимая роль, но нет необходимой функции, доступ будет запрещен.

```
// Пример
{
  path: '/trips',
  name: 'trips',
  component: () => import('../views/Trips.vue'),
  // Пользователю доступна страница рейсов,
  // если у него есть функция TripsPlane или TripsHelicopter или TripsBus
  meta: { title: 'Рейсы', authorizeFuncs: [Func.TripsPlane, Func.TripsHelicopter, Func.TripsBus] },
},
```

- **allowAnonymous** - флаг для возможности неавторизованного доступа к маршруту (если это свойство не указано, то по умолчанию неавторизованный доступ запрещен).

```
// Пример
{
  path: '/login',
  name: 'Login',
  component: () => import('../views/Login.vue'),
  // Страница входа доступна неавторизованным пользователям
  meta: { title: 'Вход - Управление перевозками', allowAnonymous: true },
},
```

Права на элемент разметки

Для удобства применения прав в прототип Vue добавлены следующие методы:

- `$hasRole(roleId: number): boolean` - проверить, есть ли у пользователя указанная роль;

```
<!-- Пример -->
<!-- Кнопка видна, если у пользователю есть роль администратора -->
<v-btn
  v-if="$hasRole($role.Admin)"
  to="/admin"
>
  Администрирование
</v-btn>
```

- `$scanRead(appFuncId: number): boolean` - проверить, доступна ли пользователю функция для чтения;

```
<!-- Пример -->
<!-- Вкладка видна, если пользователю доступна для чтения
функция TripsPlane или TripsHelicopter или TripsBus -->
<v-tab
  to="/trips"
  v-if="$scanRead($func.TripsPlane) || $scanRead($func.TripsHelicopter) || $scanRead($func.TripsBus)"
>
  Рейсы
</v-tab>
```

- `$scanWrite(appFuncId: number): boolean` - проверить, доступна ли пользователю функция для изменения.

Доступ для изменения включает в себя и доступ для чтения.

```
<!-- Пример -->
<!-- Кнопка видна, если пользователю доступна для изменения функция TripsBus -->
<v-btn
  v-if="$scanWrite($func.TripsBus)"
  color="primary"
  outlined
  to="/trips/bus/new"
```

```
>  
  Добавить  
</v-btn>
```

"Под капотом" вышеперечисленные методы обращаются к геттерам из Vuex-хранилища `user` (см файл `main.js`).

Также для избавления от дополнительных импортов в прототип Vue добавлены объекты с константами для ролей и функций - **\$func** (`const/appFunctions.js`) и **\$role** (`const/roles.js`).