

QARO - Dokumentace

QARO - Dokumentace

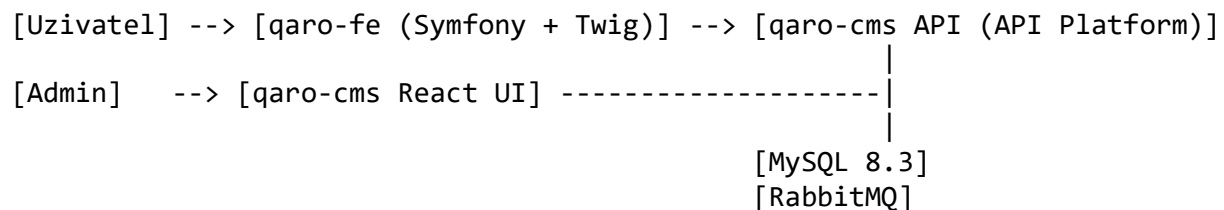
1. Úvod a architektura

Přehled systému

QARO je dvouvrstvá platforma pro spravování municipálních webů. Skládá se ze dvou hlavních aplikací:

Komponenta	Tech stack	Účel
qaro-cms	Symfony 7.0 + API Platform 3.2 (backend), React 18 + TypeScript + Vite (admin UI)	Správa obsahu, API
qaro-fe	Symfony 7.0 + Twig (šablony), Gulp (build assetů)	Veřejný web
	<ul style="list-style-type: none">Jazyk: PHP 8.3+ (backend i frontend), TypeScript/React (admin)Šablony: Twig (qaro-fe), React (qaro-cms admin)Databáze: MySQL 8.3Fronta zpráv: RabbitMQ (asynchronní úkoly přes Symfony Messenger)Autentizace: JWT (lexik/jwt-authentication-bundle)Soubory: lokální disk	

Architektura



qaro-fe konzumuje data z qaro-cms přes REST API (JWT autentizace). Admin rozhraní (React) komunikuje se stejným API.

Adresářová struktura projektu

```
qaro/
├── qaro-fe/
│   └── src/
│       ├── Controller/      # Symfony controllery
│       ├── Entity/         # Doctrine entity
│       ├── Service/        # Business logika
│       ├── Repository/     # Doctrine repository
│       ├── Security/       # Autentizace
│       ├── Twig/           # Twig extension
│       └── Command/        # CLI prikazy
```

templates/	# Twig šablony
_ template-base/	# Základní minimalní šablona
_ template1/	# Plná šablona
_ template2/	# Alternativní šablona
_ template3/	# Alternativní šablona
_ common/	# Sdílené komponenty
assets/	# Stimulus controllery, CSS (Asset Mapper)
template1/	# Gulp build konfigurace pro template1
public/	# Web root + zkompilované assety
config/	# Symfony konfigurace
migrations/	# DB migrace
translations/	# Preklady (cs/en)
qaro-cms/	# CMS (monorepo)
_ packages/	
_ backend/	# Symfony API backend
_ src/Entity/	# Doctrine entity
_ src/ApiResource/	# DTO, Procesory, Providery
_ src/Service/	# 20+ servis
_ config/	# Symfony konfig
_ migrations/	# migraci
_ frontend/	# React admin UI
_ src/modules/	# Feature moduly
_ src/api/	# Generovaný OpenAPI klient
_ vite.config.ts	# Vite konfig
_ files/	# Uložené soubory
docker/	# Docker konfigurace
docker-compose.dev.yml	# Dev prostředí
Makefile	# Dev příkazy

Jak systém načítá a zpracovává šablony

1. Požadavek přijde na qaro-fe (Symfony router)
2. Controller (DefaultController.php) určí stránku a načte data z API/DB
3. AbstractSiteController automaticky injektuje do šablony: site, breadcrumbs, searchFormWidget
4. Twig renderuje šablonu podle šablony přiřazené danému site (template1/template2/template3)
5. Twig extension poskytují data (menu, nastavení, homepage bloky, počasí, atd.)
6. Assety se linkují z /public/template1/build/ (Gulp output) (jen pro šablony 1,2,3)

2. Prostředí a instalace

Lokální vývoj (Docker)

1. Inicializace qaro-cms

cd qaro-cms

make dev-init

Vytvorí site, build image, spustí migrace

```
# 2. Spusteni
make dev-start          # Spustí všechny kontejnery
```

```
# 3. Přístup do kontejneru
make dev-in-php        # Shell do PHP-FPM
make dev-in-node       # Shell do Node
```

Lokální URL:

- `api.qaro.local` - Backend API
- `qaro.local` - CMS admin (React)
- `files.qaro.local` - File storage
- `localhost:8025` - MailHog (testování emailů)
- `localhost:15672` - RabbitMQ management

Docker služby (`docker-compose.dev.yml`): - traefik (v2.11) - Reverse proxy, routing - api (nginx) - Backend API server - php-fpm - PHP runtime - mysql (8.3) - Databáze - qaro-cms - React dev server (port 5173) - mailhog - Email testing - rabbitmq - Message queue

Deployment

Testovní uživatelé (fixtures)

Email	Role
<code>admin@mountainlift.cz</code>	<code>ROLE_ADMIN</code>
<code>superadmin@mountainlift.cz</code>	<code>ROLE_SUPER_ADMIN</code>
<code>server@localhost.cz</code>	-

3. Šablonovací systém (qaro-fe)

Šablonovací engine

Twig - PHP šablonovací engine integrovaný přes Symfony.

Konfigurace (`config/packages/twig.yaml`):

```
twig:
  file_name_pattern: '*.twig'
  paths:
    - '%kernel.project_dir%/templates'
  date:
    timezone: Europe/Prague
```

Struktura šablony

Každá šablona (template1, template2, template3) obsahuje:

```
templates/templateX/
├── base.html.twig          # Zakladni layout (HTML head, meta, CSS vars,
assety)
├── pages/                 # Jednotlive stranky
│   ├── homepage.html.twig
│   ├── news.html.twig
│   ├── news-detail.html.twig
│   ├── events.html.twig
│   ├── event-detail.html.twig
│   ├── event-calendar.html.twig
│   ├── budget.html.twig
│   ├── budget-detail.html.twig
│   ├── office-board.html.twig
│   ├── office-board-detail.html.twig
│   ├── place-detail.html.twig
│   ├── global-search.html.twig
│   ├── sitemap.html.twig
│   ├── accessibility-statement.html.twig
│   ├── 404.html.twig
│   └── unsubscribe.html.twig
├── pageComponents/      # Znovupouzitelne komponenty
│   ├── header.html.twig  # Hlavicka (250 radku)
│   ├── footer.html.twig  # Paticka (119 radku)
│   ├── breadcrumbs.html.twig # Drobeckova navigace
│   ├── pagination.html.twig # Strankovani
│   ├── search.html.twig  # Vyhledavani
│   ├── modal.html.twig   # Modalni okno
│   └── homepage/        # Homepage bloky
│       ├── fastLinks.html.twig
│       ├── latestInfo.html.twig
│       ├── mayor.html.twig
│       ├── places.html.twig
│       ├── contact.html.twig
│       ├── contactForm.html.twig
│       ├── newsletter.html.twig
│       └── weather.html.twig
└── assets/              # SCSS a JS zdrojove soubory
    ├── css/styles.scss
    ├── js/**/*.*.js
    └── img/
```

Dědičnost šablon

```
template-base/base.html.twig          # Minimalni HTML5 kostra (9 radku)
templateX/base.html.twig              # Plny layout s meta tagy, CSS vars, assety
├── templateX/pages/*.html.twig      # Konkretni stranky
```

template-base/base.html.twig definuje bloky:

- {% block title %} - Titulek stránky
- {% block header %} - Hlavička
- {% block body %} - Obsah
- {% block footer %} - Patička

templateX/base.html.twig rozšiřuje base a přidává:

- Meta tagy (og:, robots, viewport)
- Dynamický favicon a logo z nastavení site
- CSS proměnné pro barvy a fonty (konfigurovatelné z CMS admin)
- Reference na zkompilevané assety – Pred loaded fonty

Příklad CSS proměnných v base layoutu:

```
{% if siteTemplateColors.main is defined %}
  --color-main: {{ siteTemplateColors.main.hex }}!important;
  --color-main-rgb: {{ rgbObj.r }}, {{ rgbObj.g }}, {{ rgbObj.b }};
  --color-main-h: {{ hslObj.h }};
  --color-main-s: {{ hslObj.s }}%;
  --color-main-l: {{ hslObj.l }}%;
{% endif %}
```

Dostupné proměnné v šablonách

Automaticky injektované (*AbstractSiteController*)

Proměnná	Typ	Popis
site	Entity	Aktuální site (jméno, logo, favicon, lokalizace)
breadcrumbs	array	Drobečková navigace
searchFormWidget	FormView	Formulář globálního vyhledávání

Dostupné přes Twig extension

Funkce	Návratová hodnota	Popis
site_detail()	SiteDetailModel	Detail aktuálního site
site_settings()	object	Nastavení (adresa, dataBox, starosta, soc. sítě)
site_logo()	string	URL loga
site_template_settings()	array	Nastavení šablony (barvy, fonty)
template_settings()	TemplateSettingModel	Model nastavení šablony

Funkce	Návratová hodnota	Popis
main_menu_links()	array	Položky hlavního menu
fast_links()	array	Rychlé odkazy
footer_links()	array	Odkazy v patičce
homepage_settings()	HomepageSettingsModel	Konfigurace homepage bloků
homepage_latestInfo_block()	object	Novinky, události, úřední deska
homepage_places_block()	array	Místa pro homepage
homepage_weather_block()	object	Počasí
homepage_mayor_block()	object	Info o starostovi
homepage_events_block()	array	Nadcházející události (4)
homepage_news_block()	array	Poslední aktuality (3)
git_version()	string	Git commit hash (cache busting)
siteBanner(type)	object	Náhodný banner daného typu

Vlastní Twig filtry a funkce

Filtry

Filtr	Popis	Příklad
format_phone	Formátuje telefon: +CCC CCC CCC CCC	{{ phone\ format_phone }}
camelCase	Převede na camelCase	{{ 'hello-world'\ camelCase }}

Funkce

Funkce	Parametry	Popis
formatFileSize(bytes, decimals)	bytes: int, decimals: int (def. 2)	Lidsky čitelná velikost souboru
base64_svg(filePath)	filePath: string	SVG jako data URI
content_getSignPostPages(page, lang)	PageDetailModel, string	Dětské stránky pro navigaci
localization_redirect(language)	language: string	URL pro změnu jazyka
link(route)	RouteModel	Generuje link s href a target
full_link(link)	string	Absolutní URL s protokolem
generate_link(pathName)	string	Link z názvu routy
localized_path(pathName, params)	string, array	Cesta s lokálem

Funkce	Parametry	Popis
localized_url(pathName, params)	string, array	Absolutní URL s lokálem
preparePagination(paginated)	PaginatedModel	{havePrev, items, haveNext}
createNewsletterForm(options)	array	Newsletter formulář
translatable_override(key, locale, default)	string, string, string	Překlad s override
get_site_accessibility_statement()	-	Text prohlášení o přístupnosti
get_site_privacy_statement()	-	Text GDPR

Registrace šablony / Jak vytvořit novou šablonu

1. Vytvořte adresářovou strukturu:

```

templates/template-nova/
├── base.html.twig
├── pages/
│   ├── homepage.html.twig
│   ├── news.html.twig
│   └── ...
└── pageComponents/
    ├── header.html.twig
    └── footer.html.twig

```

2. Připravte konfiguraci, pro build assetu, nebo využijte stávající:
3. Registrujte v CMS:
 - V qaro-cms admin přiřadte novou šablonu k Site entitě
 - Šablona se vybírá v nastavení site
4. Zkompilujte assety:

```

cd template-nova
npm install
npx gulp deploy      # Produkcí build do public/template-nova/build/

```

Kompletní postup včetně CMS kroků viz [DOKUMENTACE-NOVA-SABLONA.md](#)

4. Assety (CSS, JS, obrázky)

Kam patří statické soubory

QARO-FE používá dva systémy pro správu assetů:

1. Gulp (šablonové assety)

Zdrojové soubory: templates/templateX/assets/

```
templates/template1/assets/
├── css/
│   └── styles.scss          # Hlavni SCSS
├── js/
│   └── **/*.js             # JavaScript moduly
├── img/
│   ├── *.{gif,jpg,png,webp} # Obrazky
│   └── icon-*.svg          # SVG ikony pro sprite
```

Výsledek buildu: public/templateX/build/

```
public/template1/build/
├── css/styles.css          # Zkompilovane CSS
├── js/scripts.js          # Spojeny a minifikovany JS
└── img/                   # Optimalizovane obrazky + SVG sprite
```

2. Symfony Asset Mapper (aplikační assety)

```
assets/
├── app.js                  # Vstupni bod
├── bootstrap.js           # Bootstrap konfigurace
├── styles/app.css         # Globalni styly
├── controllers/           # Stimulus controllery
│   ├── hello_controller.js
│   └── controllers.json
└── vendor/                # Treti strany (Turbo, Stimulus)
```

Konfigurace (config/packages/asset_mapper.yaml):

```
framework:
  asset_mapper:
    paths:
      - assets/
```

ES moduly (importmap.php):

```
return [
  'app' => ['path' => './assets/app.js', 'entrypoint' => true],
  '@hotwired/stimulus' => ['version' => '3.2.2'],
  '@symfony/stimulus-bundle' => ['path' => '...loader.js'],
  '@hotwired/turbo' => ['version' => '7.3.0'],
];
```

Build proces (Gulp)

Konfigurace: template1/gulpfile.js (516 řádků)

CSS pipeline: 1. SCSS kompilace (sass + glob) 2. PostCSS (autoprefixer, cssnano, flexbugs, pxtorem) 3. Sourcemaps 4. CSScomb formátování 5. Stylelint validace

JS pipeline: 1. Concat všech JS souborů 2. Uglify/minifikace 3. JSHint lint 4. Prettier formátování 5. Sourcemaps

Obrázky: - Imagemin optimalizace - SVG sprite generování z icon-*.svg

Příkazy:

```
cd template1/  
npx gulp code      # Dev s BrowserSync (watch)  
npx gulp makecss   # Kompilace CSS  
npx gulp makejs    # Kompilace JS  
npx gulp makeicons # SVG sprite  
npx gulp deploy    # Produkční build  
npx gulp commit    # Pre-commit kontrola
```

BrowserSync: <http://stramberk.qaro.local:81/>

Jak se assety linkují v šabloně

```
{# Gulp-compiled assets #}  
<link rel="stylesheet" href="/template1/build/css/styles.css?v={{  
git_version() }}">  
<script src="/template1/build/js/scripts.js?v={{ git_version() }}"></script>
```

```
{# Asset Mapper (Stimulus/Turbo) #}  
{% block importmap %}{{ importmap('app') }}{% endblock %}
```

Cache busting přes `git_version()` - vrací Git commit hash.

5. Konfigurace

Konfigurační soubory (qaro-fe)

Soubor	Popis
config/packages/twig.yaml	Twig engine (cesty, timezone)
config/packages/asset_mapper.yaml	Asset Mapper cesty
config/packages/framework.yaml	Symfony framework + asset versioning
config/packages/security.yaml	Bezpečnost, firewally
config/routes.yaml	Routování
importmap.php	ES module imports (Stimulus, Turbo)
composer.json	PHP závislosti
template1/gulpfile.js	Gulp build konfig
template1/package.json	Node závislosti pro Gulp

Konfigurační soubory (qaro-cms)

Soubor	Popis
--------	-------

Soubor	Popis
packages/backend/config/packages/api_platform.yaml	API Platform (formáty, paginace, JWT)
packages/backend/config/packages/security.yaml	Bezpečnost, JWT, role
packages/backend/config/packages/doctrine.yaml	Databáze
packages/backend/.env	Env proměnné (DB, JWT, Azure, email, API klíče)
packages/frontend/vite.config.ts	Vite build (React admin)
packages/frontend/tsconfig.json	TypeScript
docker-compose.dev.yml	Docker služby
Makefile	Dev příkazy
.env.template	Vzor env proměnných

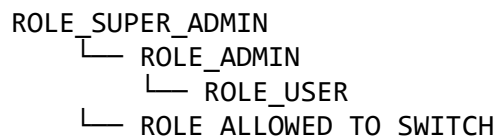
Prostředí

Env proměnné (backend):

Proměnná	Účel
DATABASE_URL	MySQL připojení
JWT_SECRET_KEY, JWT_PUBLIC_KEY, JWT_PASSPHRASE	JWT autentizace
FILES_DISK_LOCATION, FILES_PUBLIC_URL	Úložiště souborů
AZURE_CONNECTION_STRING, AZURE_CONTAINER_NAME	Azure Blob Storage
MAILER_DSN, APP_EMAIL	Emailová konfigurace
CORS_ALLOW_ORIGIN	CORS povolené domény
OPENAI_API_KEY	OpenAI integrace
GOOGLE_API_KEY, GOOGLE_CLIENT_ID	Google Analytics
OPENWEATHERMAP_API_KEY	Počasí
MESSENGER_TRANSPORT_DSN	RabbitMQ (amqp://)
QARO_AI_URL, QARO_AI_WEBHOOK_SECRET	AI webhook

Oprávnění a role

Hierarchie rolí:



Přístupová pravidla (API):

Veřejné endpointy (bez autentizace): - /api/docs - Swagger UI - /api/login_check - Přihlášení - /api/global-search - Vyhledávání - /api/reset-password, /api/request-reset-password - /api/registration - /api/google-analytics - /api/chatbot_conversations (POST) - /api/chatbot_faqs/no-pagination - /opendata

Vše ostatní vyžaduje IS_AUTHENTICATED_FULLY (JWT token).

AI funkce: - aiEnabled - Spravuje SUPER_ADMIN - aiChatbotActive - Spravuje ADMIN

6. CMS Backend (qaro-cms)

Architektura

- Symfony Messenger + RabbitMQ pro asynchronní operace
- Automatické generování API klienta pro React admin přes OpenAPI spec

Klíčové entity

Entita	Popis
Site	Kořenová entita, multi-tenant (jméno, URL, šablona, nastavení, AI)
Page	Hierarchické stránky s verzováním obsahu
News	Aktuality s plánovaným publikováním (dateStart/dateEnd)
Event	Události
OfficialBoard	Úřední deska
Budget	Rozpočet
Place	Místa
Route	Navigační položky (menu)
Campaign	Kampaně
ChatbotFaq	AI chatbot FAQ
User	Uživatelé s rolemi
File	Soubory (Azure Blob Storage)
Banner	Bannery
Setting	Nastavení site

API formáty

- JSON-LD (výchozí)
- Multipart (upload souborů)

Výchozí paginace: 10 položek na stránku (klientem nastavitelné).

7. CMS Frontend (React admin)

Tech stack

Technologie	Verze	Účel
React	18	UI framework
TypeScript	-	Typová bezpečnost
Vite	5.2	Build nástroj
Ant Design	5.18	UI komponenty
Zustand	4.4	State management
React Hook Form	7.47	Formuláře
TanStack React Query	5.49	Data fetching/cache
React Router	v6	Routing
i18next	-	Překlady
Styled Components	-	CSS-in-JS
TinyMCE / React Quill	-	WYSIWYG editory
Leaflet	-	Mapy
Recharts	-	Grafy

Struktura

```
packages/frontend/src/  
├── api/           # Generovaný OpenAPI klient + queries  
├── modules/      # Feature moduly (news, events, pages, budget, ...)  
├── components/   # Sdílené React komponenty  
├── context/      # React context  
├── theme/        # Tema (barvy, fonty)  
├── locale/       # i18n překlady  
├── tools/        # Utility funkce  
└── types/       # TypeScript typy
```

Generování API klienta

```
npm run codegen2 # Generuje typově bezpečné React Query hooks z OpenAPI specifikace
```

8. Referenční příručka

Twig Extension - kompletní přehled

UtilsExtension

- Filtr `format_phone` - Formátování telefonu
- Filtr `camelCase` - CamelCase konverze
- Funkce `formatFileSize(bytes, decimals=2)` → string
- Funkce `base64_svg(filePath)` → string (data URI)

ContentExtension

- Funkce `content_getSignPostPages(PageDetailModel, lang) → array`

LocalizationExtension

- Funkce `localization_redirect(language) → string (URL)`

LinksExtension

- Funkce `main_menu_links() → array (hlavní menu)`
- Funkce `fast_links() → array (rychlé odkazy)`
- Funkce `footer_links() → array (patička)`
- Funkce `link(RouteModel) → {href, target}`
- Funkce `full_link(link) → string (absolutní URL)`
- Funkce `generate_link(pathName) → string`
- Funkce `localized_path(pathName, params) → string`
- Funkce `localized_url(pathName, params) → string`
- Funkce `update_protocol(url) → string`

SiteExtension

- Funkce `site_detail() → SiteDetailModel`
- Funkce `site_settings() → object (adresa, socialSites, mayorName, dataBox)`
- Funkce `site_logo() → string (URL)`
- Funkce `site_template_settings() → array`
- Funkce `template_settings() → TemplateSettingModel`
- Funkce `translatable_override(key, locale, default) → string`
- Funkce `get_site_accessibility_statement() → string`
- Funkce `get_site_privacy_statement() → string`

FormsExtension

- Funkce `createNewsletterForm(options) → FormView`

HomepageExtension

- Funkce `homepage_settings() → HomepageSettingsModel`

HomepageBlocksExtension

- Funkce `homepage_latestInfo_block() → object (news, events, officialBoards)`
- Funkce `homepage_places_block() → array`
- Funkce `homepage_weather_block() → object`
- Funkce `homepage_mayor_block() → object`
- Funkce `homepage_events_block() → array (4 budoucí události)`
- Funkce `homepage_news_block() → array (3 poslední aktuality)`

PaginationExtension

- Funkce `preparePagination(PaginatedModel) → {havePrev, items, haveNext}`

- Funkce `preparePaginatedRouteParams(routeParams, page)` → array

BannerExtension

- Funkce `siteBanner(type)` → object (náhodný banner)

GitVersionExtension

- Funkce `git_version()` → string (commit hash)
- Funkce `git_hash()` → string (krátký hash)

Routy (qaro-fe)

Všechny routy jsou definované přes PHP atributy (`#[Route]`) v controllerech. Většina veřejných rout má lokalizovanou variantu s prefixem `/_{locale}`.

DefaultController - Veřejné stránky

Route	URL	Metoda	Popis
<code>site_homepage</code>	<code>/</code>	GET	Homepage
<code>site_homepage_localized</code>	<code>/_{locale}</code>	GET	Homepage (lokalizovaná)
<code>site_news</code>	<code>/aktuality</code>	GET	Seznam aktualit
<code>site_news_localized</code>	<code>/_{locale}/aktuality</code>	GET	Seznam aktualit (lok.)
<code>site_news_detail</code>	<code>/aktuality/{slug}</code>	GET	Detail aktuality
<code>site_news_detail_localized</code>	<code>/_{locale}/aktuality/{slug}</code>	GET	Detail aktuality (lok.)
<code>site_events_calendar</code>	<code>/kalendar-akci</code>	GET	Kalendář událostí
<code>site_events_calendar_localized</code>	<code>/_{locale}/kalendar-akci</code>	GET	Kalendář událostí (lok.)
<code>site_events_calendar_range</code>	<code>/kalendar-akci/{year}/{month}</code>	GET	Kalendář - konkrétní měsíc
<code>site_events_calendar_range_localized</code>	<code>/_{locale}/kalendar-akci/{year}/{month}</code>	GET	Kalendář - konkrétní měsíc (lok.)
<code>site_event_detail</code>	<code>/udalost/{slug}</code>	GET	Detail

Route	URL	Metoda	Popis
site_event_detail_localized	/_{locale}/udalost/{slug}	GET	události Detail události (lok.)
site_events_by_date_ajax	/kalendar-akci/events-by-date	GET	AJAX - události dle data
site_events_tab_content_ajax	/kalendar-akci/tab-content	GET	AJAX - obsah tabů kalendáře
site_office_board	/uredni-deska	GET	Úřední deska
site_office_board_localized	/_{locale}/uredni-deska	GET	Úřední deska (lok.)
site_office_board_archive	/uredni-deska/archiv	GET	Archiv úřední desky
site_office_board_archive_localized	/_{locale}/uredni-deska/archiv	GET	Archiv úřední desky (lok.)
site_office_board_detail	/uredni-deska/{slug}	GET	Detail úřední desky
site_office_board_detail_localized	/_{locale}/uredni-deska/{slug}	GET	Detail úřední desky (lok.)
site_place_detail	/misto/{slug}	GET	Detail místa
site_place_detail_localized	/_{locale}/misto/{slug}	GET	Detail místa (lok.)
site_city_budget	/rozklikavaci-rozpocet	GET	Rozpočet
site_city_budget_localized	/_{locale}/rozklikavaci-rozpocet	GET	Rozpočet (lok.)
site_city_budget_detail	/rozklikavaci-rozpocet/{slug}	GET	Detail rozpočtu
site_city_budget_detail_localized	/_{locale}/rozklikavaci-rozpocet/{slug}	GET	Detail rozpočtu (lok.)
global_search	/vysledky-vyhledavani	GET	Globální vyhledávání

Route	URL	Metoda	Popis
global_search_localized	/_{locale}/vysledky-vyhledavani	GET	Globální vyhledávání (lok.)
site_accessibility_statement	/prohlaseni-o-pristupnosti	GET	Prohlášení o přístupnosti
site_accessibility_statement_localized	/_{locale}/prohlaseni-o-pristupnosti	GET	Prohlášení o přístupnosti (lok.)
site_privacy_policy	/prohlaseni-o-ochrane-soukromi	GET	Ochrana soukromí
site_privacy_policy_localized	/_{locale}/prohlaseni-o-ochrane-soukromi	GET	Ochrana soukromí (lok.)
site_sitemap	/mapa-stranek	GET	Mapa stránek (HTML)
site_sitemap_localized	/_{locale}/mapa-stranek	GET	Mapa stránek (lok.)
site_sitemap_xml	/sitemap.xml	GET	Sitemap (XML)
rss_page	/rss	GET	RSS feed
site_page_change_locale	/change-locale/{lang}	GET	Změna jazyka
unsubscribe_newsletter	/unsubscribe/{id}	GET	Odhlášení z newsletteru
unsubscribe_newsletter_localized	/_{locale}/unsubscribe/{id}	GET	Odhlášení z newsletteru (lok.)
generate_ics_file	/generate-ics-file	POST	Generování ICS souboru (kalendář)
opendata_office_boards	/opendata/uredni-deska	GET	Open data - úřední deska
site_page_detail	/_{url}	GET	Detail CMS stránky

Route	URL	Metoda	Popis
site_page_detail_localized	/_{locale}/{url}	GET	(catch-all) Detail CMS stránky (lok., catch-all)

Poznámka: site_page_detail je catch-all routa - musí být definovaná jako poslední, zachytává všechny CMS stránky podle URL slug.

ApiController - Interní API

Route	URL	Metoda	Popis
api_cache_update	/api/cache/update	POST	Aktualizace cache
api_cache_delete_by_site	/api/cache/delete-by-site	POST	Smazání cache pro site
api_cache_update_grouped	/api/cache/update-grouped	POST	Skupinová aktualizace cache
api_users	/api/users	GET	Seznam uživatelů
api_login_check	/api/login_check	POST	JWT login check (config/routes.yaml)

ApiController - Webové formuláře

Route	URL	Metoda	Popis
api_newsletter_form	/web-api/newsletter-form	GET/POST	Newsletter formulář

SiteController - Správa site

Route	URL	Metoda	Popis
api_sites_save	/api/sites/save	POST	Uložení/vytvoření site
api_sites_delete	/api/sites/{externalId}	DELETE	Smazání site

CampaignController - Kampaně

Route	URL	Metoda	Popis
api_get_newsletter	/api/newsletter	POST	Newsletter kampaň

Modely (qaro-fe)

Modely jsou PHP třídy v src/Models/ sloužící jako DTO mezi CMS API a Twig šablonami. Většina používá statickou factory metodu fromPayload(array, ?string \$locale).

Hlavní modely

Model	Popis	Klíčové vlastnosti
SiteDetailModel	Kompletní konfigurace site	name, url, logo, favicon, template, localizations[],

Model	Popis	Klíčové vlastnosti
SiteSettingsModel	Nastavení site	emailForwardForms, newsletterListId, privacyStatement, accessibilityStatement contacts[], socialSites[], phones[], emails[], dataBox, mayorName, address, lat, lon, officeHours[], gaJs, showMap, aiEnabled, aiChatbotActive, cityOfficeAddress (AddressModel)
TemplateSettingModel	Nastavení šablony (barvy, fonty)	colors[], fonts[], images[], title, carouselHeight
PaginatedModel	Generická paginace	page, totalPages, totalItems, items[] (konstanta LIMIT = 10)
PageDetailModel	CMS stránka	id, url, name, description, ogTitle, indexRobots, content[], contentFiles[], parentUrl, parentName, children[]
HomepageSettingsModel	Konfigurace homepage bloků	components[]
LocalizationModel	Jazyk/lokalizace	id, name, code
AddressModel	Fyzická adresa	city, street, streetNumber, conscriptionNumber, zipCode + toOneLineString()
RouteModel	Navigační položka menu	id, name, targetBlank, url, routeType, position, children[] (rekurzivní)

Obsahové modely

Model	Popis	Klíčové vlastnosti
NewsRowModel	Řádek seznamu aktualit	id, slug, title, publishedAt, text, coverImageUrl
NewsDetailModel	Detail aktuality	id, slug, publishedAt, author, title, content, coverImageUrl, images[], files[]
EventRowModel	Řádek seznamu událostí	id, title, description, coverImageUrl, tag, location, repeatType, dateStart, dateEnd, enterSum, slug + itsOneDay(),

Model	Popis	Klíčové vlastnosti
EventDetailModel	Detail události	isAllDay(), displayTime() id, title, description, location, dateStart, dateEnd, enterSum, files[], coverImageUrl, currencyUnit, slug, closestEventDates[] + itsOneDay(), isAllDay()
EventCategoryRowModel	Kategorie událostí	id, title
OfficeBoardRowModel	Řádek úřední desky	id, title, description, category, datePublished, dateUnpublished
OfficeBoardDetailModel	Detail úřední desky	id, title, description, category, datePublished, dateUnpublished, referenceNumber, caseNumber, ecp, author[], files[]
OfficeBoardCategoryModel	Kategorie úřední desky	id, title
PlaceRowModel	Řádek seznamu míst	id, slug, title, description, type, image, rating, externalLink, openInNewTab
PlaceDetailModel	Detail místa	id, slug, title, description, images[], externalLink, openInNewTab

Rozpočtové modely (Models/Budget/)

Model	Popis	Klíčové vlastnosti
BudgetSummaryModel	Souhrn rozpočtu	real, edited, accepted, fulfilment, type (z BudgetTrait) + difference, name, slug, totalPercentage, categoryName, subcategoryName
BudgetDetailModel	Detail položky rozpočtu	real, edited, accepted, fulfilment, type (z BudgetTrait) + id, name, path, pageType
BudgetCategoryModel	Kategorie rozpočtu	key, name
BudgetDateModel	Období rozpočtu	label, value
BudgetTypeModel	Typ rozpočtu (příjem/výdaj)	type (konstanty INCOME, OUTCOME)
BudgetSlugModel	Navigace v rozpočtu	type, category, subcategory, subsubcategory, pageType

Model	Popis	Klíčové vlastnosti
BudgetDocumentRowModel	Dokument rozpočtu	file[], dateValidUntil
BudgetTrait	Sdílený trait finančních dat	real, edited, accepted, fulfilment, type

Doplňkové modely

Model	Popis	Klíčové vlastnosti
SiteBannerModel	Banner/carousel	id, type, fileUrl, url, openInNewTab
SiteContactModel	Kontakt	type, name, contact
SiteSocialModel	Sociální síť	name, url, icon
SiteMayorModel	Profil starosty	description, title, socialLink, socialLinkLabel, image
SiteWeatherModel	Počasí	tempMin, tempMax, icon, name
SiteTranslatableSettingsModel	Překladatelná nastavení	items[] + haveKey(key), get(key)
RssRowModel	RSS položka	type, id, title, description, datePublished, slug, link
CacheModel	Invalidace cache	siteId, type, id, method (konstanty: OFFICIAL_BOARD, NEWS, EVENT, ...; ACTION_UPDATE, ACTION_DELETE, ACTION_INSERT)
SaveSiteModel	Uložení site	name, domain, template, enabled, externalId, settings[], newsletterListId

Filtrovní modely (Models/Filter/)

Model	Popis	Klíčové vlastnosti
NewsSearchForm	Filtr aktualit	search, orderBy
EventSearchForm	Filtr událostí	search
OfficeBoardSearchForm	Filtr úřední desky	search, category (OfficeBoardCategoryModel), orderBy
BudgetSearchForm	Filtr rozpočtu	year (BudgetDateModel), orderBy
GlobalSearchForm	Globální vyhledávání	search

API modely (Models/Api/)

Model	Popis	Klíčové vlastnosti
CampaignModel	Newsletter kampaň	name, description, scheduledAt, coverImageUrl, components (events, news, files)
FilterEventsModel	Parametry filtrování událostí	site, from, to, search, categoryId, orderBy[], language, page
FilterOfficeBoardsModel	Parametry filtrování úřední desky	site, page
NewsletterSubscriptionModel	Odběr newsletteru	id, email, phone, firstName, lastName, sendSms, sendNewsletter

Jak založit novou šablonu

1. Založení šablony v CMS (databáze)

V CMS databázi je potřeba vložit nový záznam do tabulky template.

2. Registrace názvu v TemplateHelper

System normalizuje názvy šablon (z DB) na názvy adresářů v templates/. Je nutné přidat mapování na obou stranách.

qaro-fe: src/Utils/TemplateHelper.php

Přidejte mapování názvu z DB na název adresáře:

```
public static $templates = [  
    // ... existující mapování ...  
    'template nova' => 'template-nova',  
];
```

Klíč je lowercase název z DB sloupce name. Hodnota je název adresáře v templates/.

qaro-cms frontend: packages/frontend/src/tools/templateHelper.ts

Přidejte stejné mapování pro CMS admin (React):

```
const templates = {  
    // ... existující mapování ...
```

```
'template nova': 'template-nova',  
};
```

3. Vytvoření adresářové struktury šablony (qaro-fe)

Vytvořte nový adresář v qaro-fe/templates/:

```
templates/template-nova/  
├── base.html.twig           # Hlavní layout  
├── pages/                  # Jednotlivé stránky  
│   ├── homepage.html.twig  
│   ├── news.html.twig  
│   ├── news-detail.html.twig  
│   ├── events.html.twig  
│   ├── event-detail.html.twig  
│   ├── event-calendar.html.twig  
│   ├── budget.html.twig  
│   ├── budget-detail.html.twig  
│   ├── office-board.html.twig  
│   ├── office-board-detail.html.twig  
│   ├── place-detail.html.twig  
│   ├── global-search.html.twig  
│   ├── sitemap.html.twig  
│   ├── accessibility-statement.html.twig  
│   ├── 404.html.twig  
│   └── unsubscribe.html.twig  
├── pageComponents/       # Znovupoužitelné komponenty  
│   ├── header.html.twig  
│   ├── footer.html.twig  
│   ├── breadcrumbs.html.twig  
│   ├── pagination.html.twig  
│   ├── search.html.twig  
│   └── modal.html.twig  
└── homepage/            # Homepage bloky  
    ├── fastLinks.html.twig  
    ├── latestInfo.html.twig  
    ├── mayor.html.twig  
    ├── places.html.twig  
    ├── contact.html.twig  
    ├── contactForm.html.twig  
    ├── newsletter.html.twig  
    └── weather.html.twig
```

Tip: Nejrychlejší cesta je zkopírovat existující šablonu (např. template1) a upravit podle potřeby.

base.html.twig - povinné rozšíření

```
{% extends 'template-base/base.html.twig' %}

{% block title %}{{ pageTitle|default('') }}{% endblock %}

{% block header %}
    {% include 'template-nova/pageComponents/header.html.twig' %}
{% endblock %}

{% block body %}
    {# Obsah stránky #}
{% endblock %}

{% block footer %}
    {% include 'template-nova/pageComponents/footer.html.twig' %}
{% endblock %}
```

template-base/base.html.twig automaticky zajišťuje: - HTML5 kostru - Dynamické CSS proměnné pro barvy a fonty (z CMS nastavení) - Favicon a logo z nastavení site - Meta tagy

4. Přřazení šablony k site v CMS admin

Po spuštění migrace a deploynutí frontendu:

1. Přihlaste se do CMS admin
2. Přejděte do Nastavení > Šablony (Template Settings)
3. Nová šablona se zobrazí v seznamu dostupných šablon
4. Klikněte na Použít šablonu u požadované šablony
5. Klikněte na Upravit pro nastavení:
 - Barvy - hlavní, sekundární, textová, hlavička (hex/rgb/hsl)
 - Fonty - font nadpisů, font textu
 - Bannery - nahrané obrázky s pozicí
 - Překlady - název šablony pro každou lokalizaci
6. Uložte nastavení

Barvy a fonty se automaticky aplikují přes CSS proměnné v Twig base layoutu. Není potřeba nic dalšího nastavovat.