

# Qbix Platform Features: Video Links

[Chat](#), [Videoconferencing](#), [Events](#)  
[Group Rides](#), [People](#), [Payments](#),  
[Notifications](#), [Edit HTML](#), [Social Media](#),

[Why for Communities](#), [Why for People](#),  
[Qbix Apps](#), [Qbix Platform](#), [The Vision](#)

## Qbix Platform Features: Index

<b>Open Source</b>	
Q	Core functionality, pages, tools, sites, apps, plugins, config, installers, ...
Db	Database adapters, safety, code generation, models, migrations, ...
Users	Accounts, security, sessions, notifications, external apps, voting, ...
Streams	Data, access control, invites, realtime updates, offline notifications, ...
Places	Locations, areas, floors, columns, geolocation, interests, ...
Websites	SEO, advertising, announcements, articles, presentations, admins, ...
Assets	Badges, leaders, credits, payments, recurring subscriptions, currencies, ...
<b>Proprietary</b>	
Travel	Group rides, driver maps, matching, pickup, dropoff, notifications, ...
Calendars	Events, recurring, RSVP, attendance, payments, calendar integration, ...
Communities	Toolkit for community apps: profiles, conversations, interests, events, ...

# Q

**Core Functionality:** A unified framework for building both client and server side apps

**Pages:** Build standards-compliant webpages, routing URLs, load resources on demand

**Tools:** Build self-contained reusable components, front and back-end, to use anywhere

**Loading:** On demand, as required, loads all JS, CSS and template files for a tool

**Containment:** Tools can contain other tools, and the system loads everything in order

**Behaviors:** Multiple tools can be added to the same element

**Sites:** Accessible from laptop, tablet or mobile, used as a starting point for new users

**Apps:** Downloadable from app stores, with a back end that “just works” out of the box

**Plugins:** A mechanism for bundling pages, tools, config, etc. into a reusable package

**Config:** Store server-side configuration for apps and plugins, export some of it to clients

**Translation:** Support many languages upon launch, auto translation + human corrections

**Installers:** Automatic handling of versions and migrations for apps and plugins

**Bundles:** Bundle app resources in native app bundles to be downloaded at install-time

**PHP:** A complete PHP environment for building sites & apps, autoloading functionality, etc.

**Node.js:** An advanced JS environment for realtime updates, offline notifications and more.

**Client JS:** A full-featured environment for modern web & native apps that run anywhere.

**Q.Page:** Real-time loading and unloading of pages, JS, CSS with or w/o HTML5 history

**Q.Events:** Core framework for event-driven programming

**Tools:** Events for when parent or child tools are activated or removed

**Pages:** Events for when pages are loaded, unloaded, or activated

**Q.Socket:** For working with web sockets and realtime messaging

**Q.Pointer:** Methods for unifying touchscreen and non-touchscreen interfaces

**Hint:** Used for onboarding new users via subtle overlay animations and audio narration

**Q.Dialogs:** Standard interface for showing dialogs that can scroll, etc. on any platform

**Q.Audio:** Functions for playing, pausing and recording audio in various environments

**Helpers:**

**Getter:** A wrapper for functions to cut down on repetitive server calls

**Batcher:** A wrapper for functions to batch many requests into one

**Cache:** For caching data locally while throwing out old data when the cache gets full

**Tools:**

**Q/timestamp:** Shows an intelligent, updating timestamp that can say “2 days ago”, etc.

**Q/tabs:** Renders tabs that automatically work with Pages and URLs in real time, etc.

**Q/filter:** Implements an “autocomplete” type behavior that works on every platform

**Q/columns:** Shows a user-friendly, hierarchical, column-based interface on any platform

**Q/drawers:** Shows a user-friendly set of drawers that can house interface elements

**Q/expandable:** Can be used to render multiple expandable containers side-by-side

**Q/gallery:** Used to show various slowly changing, panning and zooming images

**Q/imagepicker:** Allows a user to select an image/photo, crop/zoom it, and upload

**Q/pie:** Renders a compact, circular progress bar that is completely customizable

**Q/ticker:** Used to implement various gently scrolling news tickers, vertical or horizontal

**Q/rating:** Used to implement 5 star ratings and such

**Q/paging:** Used to implement pagination, mostly on touchscreen platforms

**Q/form:** Can be placed inside a <form> tag to turn it into an AJAX form

**Q/bookmarklet:** Allows users to add a bookmarklet to their browser on any platform

# Db

**Database adapters:** A modular approach to supporting databases

**Timezone:** Handles server timezone differences

**Sharding:** Supports horizontally partitioning data, and rebalancing it without downtime

**Supports:**

**MySQL**

**Queries:** Allows building queries in PHP and Node.js, framework generates SQL code

**Safety:** Safely sanitize SQL queries by default, preventing injection attacks

**Encryption:** Data not used for index-based lookup can be encrypted at rest

**Models:** PHP and Node.js classes representing object-oriented methods for database rows

**Code generation**

**Models:** Automatically generate database models from schemas, and extend them

**Queries:** By default, queries are built in PHP or Node.js and SQL is generated

**Relations:** Easily model relations with hasOne and hasMany, and SQL is generated

**Connections:** Allows each module to separate its tables into their own database / prefix

**Migrations:** Scripts run by the installer, when incrementing the app version

**SQL:** Updates to the schemas

**PHP:** Scripts to run after schemas have updated, to process database content, etc.

**Helpers:** Functions to carry out optimized algorithms

**Rank:** Allows ranking data by a value

**InsertManyAndExecute:** Inserts many rows at once

**OnDuplicateKeyUpdate:** Updates rows if they already exist in the database

**Ranges:** Supports selecting rows by range, including and excluding endpoints

**Vectors:** Supports multiple columns in operations

**Criteria:** Primary keys and even array-based lookups can support multiple columns

**Joins:** Joins by multiple columns are supported, unlike most other database libraries,

# Users

**Accounts:** A user account can represent a person or an organization

**Register:** Using email, mobile number, social network, native app, invite, or **Qbix auth**

**Authentication:** Via passphrase, QR code, or **Qbix auth**

**Contacts:** Users can have friends, etc. Organizations can have admins, etc.

**Instant personalization:** Site can appear instantly personalized to new users upon arrival

**Instant social:** Upon arrival, users can instantly see their contacts and their content

**Security:**

**Device keys:** Private keys stored only on the user's device, used to sign as the user

**Access keys:** Derived from passwords or biometrics, to unlock device keys

**Keychain:** A blockchain-like system for a user to manage their list of personal devices

**Signing:** Sessions are signed with private keys

**Sessions:** Data stored server-side for a user's session on a public site or personal app

**File system:** Sessions are stored on the web server's file system

**Database:** Sessions are stored in server database and updated in transactions

**Expiration:** Duration varies by type of device (public computer vs private phone)

**Transfer:** Users can download official apps and instantly sign in without re-entering info

**oAuth:** Third party apps can instantly sign in after user approves them in the web app

**Signing:** Session IDs are cryptographically signed to help mitigate DDoS attacks

**Notifications:** Delivered by the major mobile platforms and browser vendors

**Management:** Request permissions from user, subscribe/unsubscribe, delete on logout

**Supports:**

**Chrome Web Push**

**Firefox Web Push**

**Safari Web Push**

**Mobile Chrome Web Push**

**Android Native Notifications**

**iOS Native Notifications**

**External Apps**

**Supports**

**Facebook apps**

**Android apps**

**iOS apps**

**Voting:** Implements one-user-one-vote systems for determining popularity of things

**Totals:** Find out the total number of votes or the total weight

**Tools:**

**Users/avatar:** Renders a user icon & display name as it should be seen by the app user

**Users/list:** Renders an endlessly scrollable list of community users, that can be filtered

**Users/pile:** Renders a pile of user icons, with a counter of how many are in the pile

**Users/labels:** Used to manage the user's contact labels

**Users/contacts:** An interface to let the user manage their own (or community's) contacts

**Users/sessions:** Shows a user all their authenticated sessions

**Users/getintouch:** Lets SPAM-resistant way to let others get in touch with the user

# Streams

**Data:** Streams are designed to provide a common abstraction to store and modify data

**Publishers:** A stream is published by a given user, whether a person or a community

**Multiple readers:** Many people can see the same data update in a consistent way

**Multiple writers:** Resolves conflicts, ordering, timing, and access control

**Access control:** Every operation with a stream is checked on server, whether it's allowed

**Read level:** Used to determine whether a user can see things in a stream

**Write level:** Used to determine whether a user can affect the stream

**Admin level:** Used to determine whether the user can invite or moderate others' actions

**Permissions:** A free-form system for permissions that don't fit into the above paradigm

**Roles:** These correspond to contact labels of the publisher of the stream

**Invites:** Users can invite others to a stream, which also facilitates instant signup

**Delivery:** Invites are delivered by email, sms, native notifications, FB notifications, etc.

**Instant social:** People who click the link on an invite get an instantly social experience

**Grant access:** People can grant access to others via invites

**Messages:** Represent changes in the stream, used by clients to reconstruct how it evolves

**Realtime updates**

**WebSockets:** Messages are sent in real-time, properly ordered and used to update the UI

**Presence:** People can see all their devices and sessions connecting/disconnecting

**Observers:** Streams can be observed by a limited number of non-users in real-time

**Offline Notifications:** Delivered by the major mobile platforms and browser vendors

**Subscriptions:** A user only gets notifications about streams they subscribed to

**Rules:** Users can manage dynamic rules to determine if notifications should be delivered

**Notifications:** Are automatically personalized for each user based on stream message

**Relations:** A standard way to define directional relationships between streams

**Lookup:** Relations are used for fast lookup of related streams by relation type or weight

**Tools:**

**Streams/related:** Allows adding/managing/removing/re-ordering related streams

**Streams/chat:** Enables people to have realtime conversations about a particular topic

**Streams/participants:** Used to show a real-time list of users participating in a stream

**Streams/interests:** Lets a user manage and subscribe to various interests

**Streams/html:** Lets people edit HTML content and save it to a database

**Streams/form:** Lets the user enter structured information and save it in the database

**Streams/basic:** Lets a user edit their basic profile information

**Streams/access:** Interface to manage access to a given stream

**Streams/subscription:** Interface to manage subscriptions and participation in a stream

**Streams/lookup:** Used for looking up a stream by type and title, and choosing it

**Streams/relate:** Lets the user look up streams and relate the current stream to them

**Streams/import:** Allows community admins to import a CSV file and auto-invite people

**Streams/preview:** Base functionality for showing a compact representation of a stream

**Streams/file/preview:** Shows a compact representation of a file, lets user upload a file

**Streams/image/preview:** Shows an image, let user upload an image with Q/imagepicker

**Streams/audio/preview:** Plays audio, let user record clips with Q/audio tool

# Places

**Locations:** Places on the globe

**Areas:** Rooms within locations

**Floors:** Vertical position of areas

**Columns:** Horizontal position of areas

**Geolocation:** Lets users automatically detect their location or find one on the map

**Nearby:** Functionality to connect people with information posted nearby

**For publishers:** Returns a set of  $N$  streams to publish near by information to

**For subscribers:** Returns set of four stream to subscribe to, for nearby information

**Helpers:**

**Distance:** Distance calculation

**Google:** Loads Google Places for autocomplete and search

**Countries:** Information on shapes and outlines of countries

**Postcodes:** Back-end database information on postcodes and their locations

**Heading:** Calculates heading on the Earth, or along a route

**Closest:** Finds closest points on a route to a given location

**Routing:** Does routing on the client side to avoid extra calls to mapping services

**Geocoding:** Converts coordinates to and from text-based addresses

**Internationalization:** Switches between km and miles

**Interests:** Helps people filter information by interests and location at the same time

**Tools:**

**Places/address:** Allows the user to start typing and select a properly formatted address

**Places/location:** Allows the user to select a location via autocomplete or nearby search

**Places/user/location:** Lets the user update their location and radius to search around

**Places/globe:** Renders an actual rotating globe with countries which you can select

# Websites

## Search Engine Optimization

**Metadata:** A page's title, keyword and description can be edited by website admins

**Permalink:** A page, such as an article, can be given a permalink by website admins

## Advertising

**Unit:** An advertising unit on a website

**Placement:** A way of grouping advertising units under a name

**Creative:** Content for an advertisement

**Campaign:** An advertising campaign

**Announcements:** Admins blasting messages to people subscribed to certain streams

**Articles:** An HTML document that various users can view and edit

**Presentations:** Can be played on desktop, tablet, or mobile

**Slides:** They consist of HTML and audio, can be edited easily by collaborators

## Admins

### Tools:

**Websites/advert/campaign/preview:** Displays some advertising in a place on the site

**Websites/announcement/preview:** Allows posting an announcement to certain streams

**Websites/seo:** Lets website admins manage the title, keywords, url etc. of a page

**Websites/presentation:** A presentation with HTML slides, audio etc.

# Assets

**Badges:** Awarded to people for some achievements

**Leaders:** Leaderboards for badges and credits in a given time period and region

**Credits:** Internal system of credits, may be purchased, earned, gifted, spent, etc.

## Payments

**Confirmation**

**Receipts**

**Supports:**

**Stripe**

**Authorize.net**

**ApplePay**

**AndroidPay**

**Web PaymentRequest Standard**

## Recurring subscriptions

**Cron:** Regularly checks subscriptions and executes payments as they come up

**Currencies:** Supports currency display, symbols, and localization

## Tools:

**Assets/payment:** Standard tool for making payments

**Assets/subscription:** Standard tool for starting or managing subscriptions



# **Proprietary Plugins**

# Travel

For adding travel-related functionality to apps

**Group rides:** Allows people to offer rides and join rides as passengers

**Driver maps:** Shows driver and passengers in realtime along route, recalculates if necessary

**Matching:** Based on shared origin or destination, detour duration in time or distance,

**Pickup:** Driver can pick people up on the way, they see driver arriving on the map

**Dropoff:** Driver can drop people off at a destination, or on the way from a destination

**Notifications:** People get notification when driver started driving, driver is arriving, etc.

## Tools:

**Travel/map:** Renders a map with the real time route, driver and passenger avatars on it

**Travel/trip:** An interface for the driver or passenger to manage their trip, chat, etc.

**Travel/trips:** A compact tool that lets people find a ride or offer a ride

**Travel/trip/preview:** A compact representation of a trip, its passengers and arrival time

**Travel/trip/related:** Shows trips related to a particular stream (such as location or event)

# Calendars

Adds calendar-related functionality to apps

**Events:** Title, description, photo, etc. Matching by interests, location, or community.

**Recurring:** Allows admins to set rules for reposting recurring events

**RSVP:** Manages participants in event, allows recurring RSVPs to recurring events

**Attendance:** Admins can scan QR codes of attendees, or automatic via local wifi hotspot.

**Payments:** Implements both required payments for events, and optional donations

**Calendar Integration:** Integrates with native calendar functionality for events & reminders

**Travel Integration:** Integrates with group rides, recurring events with recurring trips, etc.

## Tools:

**Calendars/event:** Lets event admins and participants see and interact with an event

**Calendars/event/preview:** Compact representation of an event, its participants, etc.

**Calendars/event/composer:** Lets event admins create a new event, recurring or not

**Calendars/recurring:** Interface to select recurrence rules for an event, trip, etc.

# Communities:

For out-of-the box customizable apps to power one or more communities

**Scripts:** For inviting people via email or sms, and giving them roles in a community

**Switcher:** Lets people join and switch between various communities

**Onboarding:** Page showing an onboarding experience for new members

**About:** Page for each community that lets its admins can modify the HTML

**People:** Page allowing users to browse community members and search by name

**Profile:** Page showing the photos, name, interests and info of a given user

**Interests:** Page that lets community members manage their interests

**Events:** Page that lets community members explore and see all the events and activities

**Conversations:** Page that powers community forums about different interests

**Occupants:** Page to view and manage occupants of each area (for buildings, cruises etc.)

**NewEvent:** Page allowing people to create a new event or activity

**NewConversation:** Page allowing people to start a new conversation

**Schedule:** Page that shows each member the events they attended or plan to attend

**Inbox:** Page that lets each member access the conversations they are a part of

**Home:** Home page for each member of the community

**Trip:** Page that lets people plan a group ride to an event or destination