

SmartQ Developer API Guide

Last updated: May 17, 2026

Overview

The SmartQ Developer API provides token-based access to customer-facing operational data, with a narrowly scoped MatriQ AI Scheduler emergency-stop action available by explicit permission.

Base endpoint:

`https://api.smartq.tv/v1`

Every request must send:

Authorization: Bearer YOUR_API_TOKEN

Example token format:

`sqtok_XXXXXXXXXXXXXXXXXXXXXXXXXXXX.YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY`

Token scopes

A token can be granted one or more of these scopes:

- `locations:read`
- `shiftq:read`
- `sportsq:read`
- `matriq:read`
- `matriq:ai-scheduler:stop`
- `campaigns:read`
- `orderboard:read`

Module scopes are the documented standard for new integrations:

- `shiftq:read` covers ShiftQ overview, sales, labor, clock-ins, POS menu, and employee data.
- `matriq:read` covers MatriQ overview, gateway, devices, provider availability, and AI Scheduler.
- `matriq:ai-scheduler:stop` allows `POST` access to the MatriQ AI Scheduler emergency-stop action for allowed locations.
- `campaigns:read` covers campaign resources.
- `orderboard:read` reserves access to OrderBoard resources as they come online.

Scope update frequency

- `locations:read`: On demand. Discovery data is read directly from Firestore when requested.

- ``shiftq:read``: ShiftQ sales refresh every 5 minutes. Labor snapshots and clock-in exceptions refresh every 5 minutes. POS stock overlay refreshes every 5 minutes. Scheduled syncs patch changed menus, groups, subgroups, and items into the existing base snapshot when POS metadata indicates updates. Full base rebuilds are reserved for initial sync or a forced manual sync. Because ``posMenu.json`` is stored as a single object, an accepted merged base change still rewrites the stored object with the updated snapshot. Employee collections typically refresh every 5 to 15 minutes.

- ``sportsq:read``: Every 5 minutes.

- ``matriq:read``: Gateway and device data refresh on heartbeat or device check-in. Provider availability refreshes when the MatriQ gateway uploads a new provider artifact for the active business day. AI Scheduler status and published drafts refresh on publish, draft change, stop, and emergency-stop events.

- ``matriq:ai-scheduler:stop``: Immediate command queue action. Resulting AI Scheduler status syncs when the gateway applies the stop event.

- ``campaigns:read``: On demand.

- ``orderboard:read``: Coming soon.

Full examples by scope

Full Developer API examples are also published at:

- ``https://developer.smartq.tv/examples``

These examples are string-format references. Placeholder values such as ``GROUP_ID``, ``LOCATION_ID``, ``ISO_TIMESTAMP``, ``STATUS_STRING``, and ``NUMBER`` are intentional.

```
### `locations:read`
```

```
{
  "ok": true,
  "version": "v1",
  "resource": "locations",
  "groupId": "GROUP_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "token": {
    "scopeType": "group_or_location",
    "scopes": ["locations:read", "shiftq:read", "matriq:read"],
    "rateLimitPerMinute": "NUMBER",
    "expiresAt": "ISO_TIMESTAMP_OR_NULL",
    "status": "active_or_revoked_or_expired",
    "allowedLocationIds": ["LOCATION_ID_IF_LOCATION_SCOPED"],
    "tokenPrefix": "TOKEN_PREFIX"
  },
}
```

```

"data": {
  "refresh": {
    "cadence": "CADENCE_STRING",
    "description": "REFRESH_DESCRIPTION"
  },
  "locations": [
    {
      "id": "LOCATION_ID",
      "name": "LOCATION_NAME",
      "locationShortName": "LOCATION_SHORT_NAME",
      "timezone": "IANA_TIMEZONE",
      "slug": "LOCATION_SLUG",
      "status": "STATUS_STRING",
      "modulesEnabled": {
        "shiftq": "BOOLEAN",
        "sportsq": "BOOLEAN",
        "matriq": "BOOLEAN",
        "campaigns": "BOOLEAN",
        "orderboard": "BOOLEAN"
      },
      "resources": ["RESOURCE_NAME"],
      "endpoints": {
        "root": "https://api.smartq.tv/v1/GROUP_ID/LOCATION_ID",
        "modules": "https://api.smartq.tv/v1/GROUP_ID/LOCATION_ID/modules",
        "shiftq": "https://api.smartq.tv/v1/GROUP_ID/LOCATION_ID/shiftq",
        "sportsq": "https://api.smartq.tv/v1/GROUP_ID/LOCATION_ID/sportsq",
        "matriq": "https://api.smartq.tv/v1/GROUP_ID/LOCATION_ID/matriq"
      }
    }
  ]
}

```

```

### `shiftq:read`

```

```

{

```

```

"ok": true,
"version": "v1",
"resource": "shiftq",
"groupId": "GROUP_ID",
"locationId": "LOCATION_ID",
"generatedAt": "ISO_TIMESTAMP",
"data": {
  "location": {
    "id": "LOCATION_ID",
    "name": "LOCATION_NAME",
    "timezone": "IANA_TIMEZONE"
  },
  "refresh": {
    "sales": { "cadence": "CADENCE_STRING", "description": "SALES_REFRESH_DESCRIPTION" },
    "labor": { "cadence": "CADENCE_STRING", "description": "LABOR_REFRESH_DESCRIPTION" },
    "posMenu": { "cadence": "CADENCE_STRING", "description":
"POS_MENU_REFRESH_DESCRIPTION" },
    "clockIns": { "cadence": "CADENCE_STRING", "description":
"CLOCK_IN_REFRESH_DESCRIPTION" },
    "employees": { "cadence": "CADENCE_STRING", "description":
"EMPLOYEE_REFRESH_DESCRIPTION" }
  },
  "summary": {
    "businessDate": "YYYYMMDD",
    "salesUpdatedAt": "ISO_TIMESTAMP",
    "laborUpdatedAt": "ISO_TIMESTAMP",
    "top5Leaderboard": [
      { "employeeId": "EMPLOYEE_ID", "employeeName": "EMPLOYEE_NAME", "metric":
"METRIC_NAME", "value": "NUMBER", "rank": "NUMBER" }
    ],
    "clockedInCount": "NUMBER",
    "soldOutItemCount": "NUMBER"
  }
}
}

```

Additional `shiftq:read` POS menu example:

```
{
  "ok": true,
  "version": "v1",
  "resource": "shiftq/pos-menu",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "location": {
      "id": "LOCATION_ID",
      "name": "LOCATION_NAME",
      "timezone": "IANA_TIMEZONE"
    },
    "refresh": {
      "cadence": "CADENCE_STRING",
      "description": "POS_MENU_REFRESH_DESCRIPTION"
    },
    "syncStatus": {
      "lastStatus": "STATUS_STRING",
      "lastCheckedAt": "ISO_TIMESTAMP",
      "lastSyncedAt": "ISO_TIMESTAMP",
      "fullMenuRewrite": "POLICY_STRING",
      "incrementalBaseUpdates": "POLICY_STRING",
      "stockOverlayRefresh": "CADENCE_STRING"
    },
    "menuMetadata": {
      "restaurantGuid": "POS_RESTAURANT_GUID",
      "lastUpdated": "ISO_TIMESTAMP"
    },
    "menuSnapshot": {
      "lastUpdated": "ISO_TIMESTAMP",
      "lastSynced": "ISO_TIMESTAMP"
    }
  }
}
```

```
"soldOutItems": {
  "count": "NUMBER",
  "items": [
    { "key": "LOCATION_ID:INDEX:ITEM_NAME", "name": "ITEM_NAME" }
  ]
}
}
```

```
### `sportsq:read`
```

```
{
  "ok": true,
  "version": "v1",
  "resource": "sportsq",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "location": {
      "id": "LOCATION_ID",
      "name": "LOCATION_NAME",
      "timezone": "IANA_TIMEZONE"
    },
    "effectiveDate": "YYYY-MM-DD",
    "refresh": {
      "cadence": "CADENCE_STRING"
    },
    "guide": [
      {
        "eventId": "EVENT_ID",
        "title": "EVENT_TITLE",
        "league": "LEAGUE_NAME",
        "homeTeam": "HOME_TEAM_NAME",
        "awayTeam": "AWAY_TEAM_NAME"
      }
    ]
  }
}
```

```

    ]
  }
}

### `matriq:read`

{
  "ok": true,
  "version": "v1",
  "resource": "matriq",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "location": {
      "id": "LOCATION_ID",
      "name": "LOCATION_NAME",
      "timezone": "IANA_TIMEZONE"
    },
    "refresh": {
      "gateway": { "cadence": "CADENCE_STRING" },
      "devices": { "cadence": "CADENCE_STRING" },
      "providerAvailable": { "cadence": "CADENCE_STRING", "description":
"PROVIDER_REFRESH_DESCRIPTION" },
      "aiScheduler": { "cadence": "CADENCE_STRING", "description":
"AI_SCHEDULER_REFRESH_DESCRIPTION" }
    },
    "summary": {
      "deviceIds": ["DEVICE_ID"],
      "providerCount": "NUMBER",
      "artifactCount": "NUMBER",
      "latestProviderArtifactUpdatedAt": "ISO_TIMESTAMP",
      "aiSchedulerPublished": "BOOLEAN",
      "aiSchedulerRunning": "BOOLEAN"
    }
  }
}

```

```
}
```

Additional `matriq:read` provider availability example:

```
{  
  "ok": true,  
  "version": "v1",  
  "resource": "providerAvailable",  
  "groupId": "GROUP_ID",  
  "locationId": "LOCATION_ID",  
  "generatedAt": "ISO_TIMESTAMP",  
  "data": {  
    "location": {  
      "id": "LOCATION_ID",  
      "name": "LOCATION_NAME",  
      "timezone": "IANA_TIMEZONE"  
    },  
    "deviceIds": ["DEVICE_ID"],  
    "businessDate": {  
      "active": "YYYYMMDD",  
      "calendarDate": "YYYY-MM-DD",  
      "timezone": "IANA_TIMEZONE"  
    },  
    "providers": [  
      {  
        "providerKey": "PROVIDER_KEY",  
        "providerLabel": "PROVIDER_LABEL",  
        "providerLogo": "PROVIDER_LOGO_URL"  
      }  
    ],  
    "artifacts": [  
      {  
        "source": "ARTIFACT_SOURCE",  
        "deviceId": "DEVICE_ID",  
        "path": "BUCKET_PATH/provider-available-YYYY-MM-DD.json",  
        "updatedAt": "ISO_TIMESTAMP"  
      }  
    ]  
  }  
}
```

```

    }
  ],
  "providersAvailable": [
    {
      "eventId": "EVENT_ID",
      "title": "EVENT_TITLE",
      "providerLabel": "PROVIDER_LABEL",
      "providerLogo": "PROVIDER_LOGO_URL",
      "homeLogo": "HOME_LOGO_URL",
      "awayLogo": "AWAY_LOGO_URL"
    }
  ]
}
}

```

`matriq:ai-scheduler:stop`

```

{
  "ok": true,
  "version": "v1",
  "resource": "matriq/aiScheduler/emergency-stop",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "status": "STATUS_STRING",
    "deviceId": "DEVICE_ID",
    "commandId": "COMMAND_ID",
    "reason": "REASON_STRING"
  }
}

```

`campaigns:read`

```

{
  "ok": true,
  "version": "v1",

```

```
"resource": "campaigns",
"groupId": "GROUP_ID",
"locationId": "LOCATION_ID",
"generatedAt": "ISO_TIMESTAMP",
"data": {
  "refresh": {
    "cadence": "CADENCE_STRING"
  },
  "campaigns": [
    {
      "id": "CAMPAIGN_ID",
      "name": "CAMPAIGN_NAME",
      "startsAt": "ISO_TIMESTAMP",
      "endsAt": "ISO_TIMESTAMP",
      "status": "STATUS_STRING"
    }
  ]
}
```

```
### `orderboard:read`
```

```
{
  "ok": true,
  "version": "v1",
  "resource": "orderboard",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "refresh": {
      "cadence": "CADENCE_STRING"
    },
    "status": "STATUS_STRING",
    "message": "MESSAGE_STRING"
  }
}
```

```
}
```

Recommended request flow

1. Call the locations endpoint first.
2. Use the location endpoint or modules endpoint to discover enabled modules and granted resources.
3. Call the module or subresource endpoint you need.
4. Keep the token on your backend only.

Refresh metadata

Most responses now include a `refresh` block so developers can see the expected update cadence without reading separate documentation.

Business-day boundary:

- SmartQ sales and ShiftQ business-date resources follow the POS business day boundary of `4:01 AM` through `4:00 AM` local time.
- Example: at `YYYY-MM-DD 3:30 AM` local time, the active business date remains `YYYYMMDD`.

Example:

```
{
  "refresh": {
    "cadence": "Every 5 minutes",
    "description": "POS stock availability refreshes every 5 minutes. Scheduled syncs patch changed menus, groups, subgroups, and items into the existing base snapshot when POS metadata indicates updates. A full base snapshot rebuild is reserved for initial sync or a manual forced sync."
  }
}
```

Discovery endpoints

List accessible locations

```
GET /v1/GROUP_ID/locations
```

Returns:

- location identity
- enabled modules
- granted endpoints for each location
- refresh metadata

Discover one location

GET /v1/GROUP_ID/LOCATION_ID

Returns:

- enabled modules for that location
- granted endpoints for that location

Group modules summary

GET /v1/GROUP_ID/modules

Returns:

- each accessible location
- enabled module flags
- granted endpoints filtered to enabled modules

Location modules summary

GET /v1/GROUP_ID/LOCATION_ID/modules

Returns:

- group summary
- location summary
- enabled module flags
- granted endpoints filtered to enabled modules

ShiftQ endpoints

ShiftQ overview

GET /v1/GROUP_ID/LOCATION_ID/shiftq

Returns:

- location summary
- refresh policy by ShiftQ data type
- enabled modules
- data availability flags
- top 5 leaderboard
- clock-in exception counts
- sold-out item count
- manager-on-duty and labor role summary
- endpoints for ShiftQ subresources

ShiftQ sales

GET /v1/GROUP_ID/LOCATION_ID/shiftq/sales

Returns:

- sales snapshot
- analytics summary
- labor snapshot
- top 5 leaderboard
- refresh metadata

ShiftQ labor

GET /v1/GROUP_ID/LOCATION_ID/shiftq/labor

Returns:

- labor snapshot
- computed labor details
- M.O.D. rows
- scheduled M.O.D. rows
- job code clock-in counts
- refresh metadata

ShiftQ clock-ins

GET /v1/GROUP_ID/LOCATION_ID/shiftq/clock-ins

Returns:

- clocked-in count
- wrong job code clock-ins
- scheduled but missing
- clocked in but not scheduled
- early clock-ins
- late clock-ins
- role coverage summary
- refresh metadata

ShiftQ POS menu

GET /v1/GROUP_ID/LOCATION_ID/shiftq/pos-menu

Returns:

- menu sync status
- menu metadata

- menu snapshot timestamps
- sold-out item summary
- explicit full-menu rewrite policy

Important behavior:

- The full POS menu snapshot is not rewritten every 5 minutes.
- The full `posMenu.json` base snapshot is not rebuilt every 5 minutes.
- Initial sync or a manual forced sync can rebuild the base snapshot.
- Scheduled sync runs refresh the stock overlay every cycle.
- When POS metadata indicates updates, scheduled syncs patch changed menus, groups, subgroups, and items into the existing base snapshot instead of treating every update as a forced full rebuild.
- Because `posMenu.json` is stored as one object, any accepted merged base change still results in that stored object being rewritten with the updated snapshot.

ShiftQ employees

```
GET /v1/GROUP_ID/LOCATION_ID/shiftq/employees
```

Returns:

- employees
- jobs
- shifts
- time entries
- labor snapshot
- refresh metadata

SportsQ endpoint

SportsQ guide

```
GET /v1/GROUP_ID/LOCATION_ID/sportsq
```

Optional query parameter:

- `date=YYYY-MM-DD`

Returns:

- location summary
- refresh metadata
- effective date
- SportsQ settings
- filtered guide payload

MatriQ endpoints

MatriQ overview

GET /v1/GROUP_ID/LOCATION_ID/matriq

Returns:

- device count
- gateway summary
- latest heartbeat summary
- endpoints for gateway, devices, provider availability, and AI Scheduler
- provider summary
- AI Scheduler summary
- AI Scheduler emergency-stop action metadata when the token includes `matriq:ai-scheduler:stop`

MatriQ gateway health

GET /v1/GROUP_ID/LOCATION_ID/matriq/gateway

Returns:

- paired gateway document
- latest paired device heartbeat rows
- refresh metadata

MatriQ devices

GET /v1/GROUP_ID/LOCATION_ID/matriq/devices

Returns:

- paired MatriQ device rows
- status
- last heartbeat timestamps
- health payloads
- refresh metadata

MatriQ provider availability

GET /v1/GROUP_ID/LOCATION_ID/matriq/providerAvailable

Preferred direct alias:

- `GET /v1/GROUP_ID/LOCATION_ID/providerAvailable`

Returns:

- device IDs
- provider catalog with provider logo URLs
- SportsQ provider settings
- TV guide settings
- device artifacts
- latest published artifact
- provider logo URLs on each provider-matched event and match row
- SportsQ logo enrichment for events, teams, and competitors when available
- refresh metadata

MatriQ AI Scheduler

GET /v1/GROUP_ID/LOCATION_ID/matriq/aiScheduler

Returns:

- device IDs used for the synced status lookup
- current AI Scheduler runtime status
- current published business-day draft when one is published
- current business day string
- status sync timestamp
- published draft sync timestamp
- available action metadata, including the emergency-stop `POST` endpoint and required scope
- refresh metadata

MatriQ AI Scheduler emergency stop

POST /v1/GROUP_ID/LOCATION_ID/matriq/aiScheduler/emergency-stop

Required scope:

- `matriq:ai-scheduler:stop`

Request body:

```
{
  "reason": "REASON_STRING"
}
```

Returns:

- queued action status
- MatriQ device ID targeted by the command
- queued command ID
- request timestamp

- refresh metadata for the AI Scheduler status feed

Notes:

- This action queues a command for the paired MatriQ gateway and returns `202 Accepted` when queued successfully.
- The MatriQ gateway applies the emergency stop locally, syncs the updated AI Scheduler status to Firestore and bucket storage, and the `GET /matriq/aiScheduler` endpoint reflects the new stop state after that stop event sync is written.

Campaigns endpoints

Group campaigns

GET /v1/GROUP_ID/campaigns

Returns:

- group-level TV campaigns
- group-level guest access campaigns
- per-location campaign endpoints
- refresh metadata

Location campaigns

GET /v1/GROUP_ID/LOCATION_ID/campaigns

Returns:

- location TV campaigns
- campaign asset rows
- guest access campaign rows
- campaign device status summary
- refresh metadata

OrderBoard endpoint

OrderBoard status

GET /v1/GROUP_ID/LOCATION_ID/orderboard

or

GET /v1/GROUP_ID/orderboard

Returns:

- status string
- message that the OrderBoard Developer API is coming soon

- refresh metadata

Example response shape

```
{
  "ok": true,
  "version": "v1",
  "resource": "shiftq/clock-ins",
  "groupId": "GROUP_ID",
  "locationId": "LOCATION_ID",
  "generatedAt": "ISO_TIMESTAMP",
  "data": {
    "location": {
      "id": "LOCATION_ID",
      "name": "LOCATION_NAME",
      "timezone": "IANA_TIMEZONE"
    },
    "refresh": {
      "cadence": "CADENCE_STRING",
      "description": "REFRESH_DESCRIPTION"
    },
    "clockedInCount": "NUMBER",
    "exceptions": {
      "wrongJobCodeClockIns": [],
      "scheduledButMissing": [],
      "clockedInButNotScheduled": [],
      "earlyClockIns": [],
      "lateClockIns": []
    }
  }
}
```

Common HTTP responses

- `200 OK`: request succeeded
- `401 Unauthorized`: token missing or invalid
- `403 Forbidden`: token is revoked, expired, outside allowed locations, or missing the required scope

- `404 Not Found`: group, location, or resource was not found
- `429 Too Many Requests`: token rate limit exceeded

Security guidance

- Keep SmartQ tokens on your backend only.
- Do not embed tokens in browser JavaScript or mobile apps.
- Rotate tokens if they are shared too broadly.
- Revoke tokens immediately if they are exposed.

Support

If a customer needs a narrower export shape, a custom field map, or a longer-term contract for endpoint stability, SmartQ can extend the API without exposing internal credentials or admin-only data.

Maintenance requirement

When API scopes, endpoint names, or module structure change, update all of the following in the same change set:

- `developer.smartq.tv`
- `docs/SmartQ-Customer-API.postman_collection.json`
- `docs/SmartQ-Customer-API.postman_environment.json`
- `/admin/internal/customer-api/`
- `docs/smartq-customer-api-guide.md`
- `docs/SmartQ-Customer-API-Guide.pdf`