API Endpoints

Movies

GET `/api/v1/movies` (list, filter + paging)

POST `/api/v1/movies` *(manager)* create

GET `/api/v1/movies/{movieId}` details

PATCH `/api/v1/movies/{movieId}` *(manager)* update fields

DELETE `/api/v1/movies/{movieId}` *(manager)* archive/delete

———

Movie links

GET `/api/v1/movies/{movieId}/images`

POST `/api/v1/movies/{movieId}/images` *(manager)* add image/link

PATCH `/api/v1/movies/{movieId}/images/{imageId}` *(manager)* update (type, sortOrder, url)

DELETE `/api/v1/movies/{movieId}/images/{imageId}` *(manager)* remove

———

Cinemas

GET `/api/v1/cinemas`

POST `/api/v1/cinemas` *(manager)* create

GET `/api/v1/cinemas/{cinemaId}`

PATCH `/api/v1/cinemas/{cinemaId}` *(manager)* update

DELETE `/api/v1/cinemas/{cinemaId}` *(manager)* archive/delete

———

Halls

GET `/api/v1/cinemas/{cinemaId}/halls`

POST `/api/v1/cinemas/{cinemaId}/halls` *(manager)* create

GET `/api/v1/halls/{hallId}`

PATCH `/api/v1/halls/{hallId}` *(manager)* update (name, hallType, status)

DELETE `/api/v1/halls/{hallId}` *(manager)* archive/delete

———

Seats

GET `/api/v1/halls/{hallId}/seats`

POST `/api/v1/halls/{hallId}/seats` *(manager)* add seat

POST `/api/v1/halls/{hallId}/seats/bulk` *(manager)* bulk create layout

PATCH `/api/v1/halls/{hallId}/seats/{seatId}` *(manager)* update
(row/number/type/status)

DELETE `/api/v1/halls/{hallId}/seats/{seatId}` *(manager)* remove

———

Screenings

Browse/search

GET `/api/v1/screenings` (filters: dateFrom/dateTo, cinemaId, movieId)

GET `/api/v1/screenings/{screeningId}`

GET `/api/v1/screenings/{screeningId}/seatmap` (Seat + ScreeningSeat status)

Create/manage

POST `/api/v1/screenings` *(manager)* create (movieId, hallId, startAt, endAt, pricing)

PATCH `/api/v1/screenings/{screeningId}` *(manager)* update (time/status)

DELETE `/api/v1/screenings/{screeningId}` *(manager)* cancel/delete

Screening seat control (BLOCK/UNBLOCK etc)

POST `/api/v1/screenings/{screeningId}/seats/{seatId}/block` *(manager/staff)*

POST `/api/v1/screenings/{screeningId}/seats/{seatId}/unblock` *(manager/staff)*

PATCH `/api/v1/screenings/{screeningId}/seats/{seatId}` *(manager/staff)* update status (AVAILABLE/BLOCKED)

GET `/api/v1/screenings/{screeningId}/seats` *(manager/staff)* list with statuses

—--

Orders

Customer

POST `/api/v1/orders` *(customer)* create order from hold

GET `/api/v1/me/orders` *(customer)* list my orders

GET `/api/v1/me/orders/{orderId}` *(customer)* order details

PATCH `/api/v1/me/orders/{orderId}` *(customer)* update billing info / contact info

DELETE `/api/v1/me/orders/{orderId}` *(customer)* cancel (if not paid)

Manager (optional reporting / support)

GET `/api/v1/orders` *(manager)* search orders (by date, screening, status, buyer)

GET `/api/v1/orders/{orderId}` *(manager)* details

PATCH `/api/v1/orders/{orderId}` *(manager)* admin-cancel/refund status

Payments (mock or integration-ready)

POST `/api/v1/orders/{orderId}/payments` *(customer)* create payment attempt

GET `/api/v1/orders/{orderId}/payments` *(owner/manager)* list payment attempts

GET `/api/v1/payments/{paymentId}` *(owner/manager)*

PATCH `/api/v1/payments/{paymentId}` *(system/manager)* update status (SUCCEEDED/FAILED)

POST `/api/v1/orders/{orderId}/pay` *(customer)* shortcut "mock pay now" endpoint (optional)

Hold

POST `/api/v1/screenings/{screeningId}/holds` *(customer)* (seatIds → holdId, expiresAt)

GET `/api/v1/holds/{holdId}` *(owner)* (kontroll)

PATCH `/api/v1/holds/{holdId}` *(owner)* (muuda seatIds)

DELETE `/api/v1/holds/{holdId}` *(owner)* (release)

Tickets

Customer
GET `/api/v1/me/tickets` list my tickets (filters: upcoming/past)

GET `/api/v1/me/tickets/{ticketId}` ticket details

GET `/api/v1/me/tickets/{ticketId}/qr` get QR code payload/image data

POST `/api/v1/me/tickets/{ticketId}/transfer` *(optional)* transfer to another email/user

POST `/api/v1/me/tickets/{ticketId}/refund-request` *(optional)* request refund

Staff/Manager

GET `/api/v1/tickets/{ticketId}` view any ticket

PATCH `/api/v1/tickets/{ticketId}` *(manager)* force status update (admin support)

POST `/api/v1/tickets/{ticketId}/invalidate` *(manager)* invalidate ticket (fraud)

——

Validation

GET `/api/v1/validations` *(staff/manager)* filter by screening/date/result

GET `/api/v1/validations/{validationId}` *(staff/manager)*

POST `/api/v1/validations/scan` *(staff)*

POST `/api/v1/validations/manual` *(staff)*

DELETE `/api/v1/validations/{validationId}` *(manager)* remove log (rare, admin only)

——

Staff

GET `/api/v1/staff/screenings/today` *(staff)* assigned or all today

GET `/api/v1/staff/screenings/{screeningId}/entry-stats` *(staff)* validated count, remaining, alerts

GET `/api/v1/staff/screenings/{screeningId}/issues` *(optional)* incidents list

POST `/api/v1/staff/screenings/{screeningId}/issues` *(optional)* create incident

Reports

GET `/api/v1/reports/sales` *(manager)* totals by date/cinema/movie

GET `/api/v1/reports/screenings/{screeningId}` *(manager)* occupancy, revenue, validation stats

GET `/api/v1/reports/no-shows` *(manager)* sold vs validated

GET `/api/v1/reports/seats` *(manager)* seat utilization heatmap data

GET `/api/v1/screenings/{screeningId}/entry-stats`