

Complete API List for Advanced Task Management Backend

Below is a comprehensive list of REST and Socket.IO endpoints you should develop, reflecting all major and advanced features from your PRD. Each endpoint includes the HTTP method and a brief description.

1. Authentication & User Management

Endpoint	Method	Description
/api/v1/auth/register	POST	Register a new user
/api/v1/auth/login	POST	User login and JWT issuance
/api/v1/auth/oauth	POST	OAuth login (Google/GitHub)
/api/v1/auth/forgot	POST	Request password reset email
/api/v1/auth/reset	POST	Reset password
/api/v1/auth/verify	POST	Verify user email
/api/v1/users/me	GET	Get current user profile
/api/v1/users/:id	GET	Get user by ID
/api/v1/users/:id	PATCH	Update user profile
/api/v1/users/roles	PATCH	Change user roles

2. Team & Organization

Endpoint	Method	Description
/api/v1/teams	GET	List user's teams
/api/v1/teams	POST	Create a team
/api/v1/teams/:id	GET	Get team details
/api/v1/teams/:id	PATCH	Update team info
/api/v1/teams/:id	DELETE	Delete a team
/api/v1/teams/:id/members	GET	List team members
/api/v1/teams/:id/invite	POST	Invite new member via email
/api/v1/teams/:id/roles	PATCH	Update member roles in team
/api/v1/teams/:id/leave	POST	Leave a team

3. Projects, Boards, Lists, and Tasks

Endpoint	Method	Description
/api/v1/projects	GET	List projects (by team/user)
/api/v1/projects	POST	Create new project
/api/v1/projects/:id	GET	Get project details
/api/v1/projects/:id	PATCH	Update project info
/api/v1/projects/:id	DELETE	Delete project
/api/v1/boards	POST	Create board under a project
/api/v1/boards/:id	GET	Get board details
/api/v1/boards/:id	PATCH	Update board
/api/v1/boards/:id	DELETE	Delete board
/api/v1/lists	POST	Create list on a board
/api/v1/lists/:id	PATCH	Update list
/api/v1/lists/:id	DELETE	Delete list
/api/v1/tasks	GET	List/filter/search tasks
/api/v1/tasks	POST	Create new task
/api/v1/tasks/:id	GET	Get task details
/api/v1/tasks/:id	PATCH	Update task (status, details etc.)
/api/v1/tasks/:id	DELETE	Delete task
/api/v1/tasks/:id/assign	PATCH	Assign/reassign task to a user
/api/v1/tasks/:id/attachments	POST	Upload attachment to a task
/api/v1/tasks/:id/attachments	GET	Get all attachments for a task
/api/v1/tasks/:id/attachments/:a	DELETE	Remove an attachment from a task
/api/v1/tasks/:id/labels	PATCH	Add/edit labels/tags for a task
/api/v1/tasks/:id/checklist	PATCH	Manage task checklists

4. Comments & Activity

Endpoint	Method	Description
/api/v1/tasks/:id/comments	GET	Get all comments on a task
/api/v1/tasks/:id/comments	POST	Add comment
/api/v1/tasks/:id/comments/:cid	PATCH	Edit a comment
/api/v1/tasks/:id/comments/:cid	DELETE	Delete a comment

Endpoint	Method	Description
/api/v1/activities	GET	Get recent activity feed (boards/tasks)

5. Notifications & Real-Time (Socket.IO)

Endpoint	Method/Channel	Description
/api/v1/notifications	GET	Fetch all user notifications
/api/v1/notifications	PATCH	Mark notifications as read
SOCKET: notifications	WS	Push real-time notifications
SOCKET: task-updates	WS	Push real-time task/board updates

6. Search & Filtering

Endpoint	Method	Description
/api/v1/search	GET	Search boards/tasks by keywords, filters, tags

7. Integrations & Background Jobs

Endpoint	Method	Description
/api/v1/integrations/email	POST	Trigger/invite via email
/api/v1/integrations/sms	POST	Send SMS notification (Twilio)

8. Miscellaneous

Endpoint	Method	Description
/api/v1/files/upload	POST	Generic file upload
/api/v1/files/download/:id	GET	Download a file/attachment
/health	GET	Health check endpoint
/docs	GET	Swagger/OpenAPI documentation UI

9. Admin & Settings

Endpoint	Method	Description
/api/v1/admin/users	GET	List all users (admin only)
/api/v1/admin/teams	GET	List all teams (admin only)
/api/v1/admin/settings	PATCH	Update platform/system settings

Note:

- All endpoints should follow RESTful patterns, use proper HTTP status codes, and be secured with appropriate authentication and role-based authorization.
- Add more endpoints as needed for error handling, preferences, and new features as you grow your application^{[1] [2] [3] [4]}.

This list ensures your backend covers core, advanced, integration, and admin features commonly expected in a professional project management system.



1. <https://www.geeksforgeeks.org/create-a-backend-task-management-system-using-microservices/>
2. https://github.com/ashishkumarpalai/tacnique_assignment
3. <https://github.com/dashrathshinde07/task-manager-nodejs>
4. <https://www.geeksforgeeks.org/node-js/task-management-system-using-node-and-express-js/>