

# Deploying ScheduleDB to Production

- [Summary](#)
- [History](#)
- [TSSG Management Responsibilities](#)
- [Access](#)
- [TSSG Management Input](#)
- [Conclusion](#)

## Summary

The ScheduleDB project consists of a set of micro-services that allow a user with administrator credentials to login and maintain the database collections of Teams, Users, Venues and Meeting Schedules. The user facing services include:

- frontend : a User Interface for login and data management.
- website: a micro-service implementation of the current tssg.tech website. Includes a schedule page that automatically updates with the next three General Meeting schedules.

## History

For the last few years that tssg.tech has had a page to list the upcoming schedules: a developer, like [Joel Sharasheff](#), had to make code changes every week to keep the page updated.

The ScheduleDB project was born out of a need to create a process whereby any authorized user could make the schedule, venues, teams and user changes into a database once. The website (tssg.tech) would then update the schedule page automatically every week.

## TSSG Management Responsibilities

Before these implemented services can be deployed to replace the [tssg.tech](#) production website, the Web Team is requesting that all TSSG Management stakeholders test the frontend and website services. Management will then update the TSSG Management Input table below for a Go or No-Go vote to Production along with Constructive Feedback.

Management will also need to decide who will update the General Meeting Schedules within the services.

## Access

Access to the development instance of the FrontEnd Service is available by navigating your browser to the development endpoints:

- **Administrative FrontEnd** - <https://frontend.sdb.technologynursery.org>
  - Try out adding users, teams, venues, and meeting schedules.
- **Proposed Website** - <https://website.sdb.technologynursery.org>
  - Click on the schedule icon to get the next three scheduled weeks from the database.

Login credentials for the administrator "admin" will be provided to the tssg-management group in an email.

**Once you have logged in as admin, please create an administrative user for yourself to do the testing. Log out as admin and then login as your newly created user.**

## TSSG Management Input

Please update this table with your Vote and/or Feedback.

Manager	Team	Vote	Comments / Constructive Feedback
		<ul style="list-style-type: none"><li>• Go</li></ul>	
		or	
		<ul style="list-style-type: none"><li>• No</li></ul>	
		-	
		Go	

Shawn Flynn	TSSG	✓Go	<p>Found the functionality adequate for the current tasks required.</p> <p>Future:</p> <ul style="list-style-type: none"> <li>• Copy/Paste meeting info (or a duplicate function)</li> <li>• Schedule multiple dates at same venue</li> <li>• Ability to store comment data: <ul style="list-style-type: none"> <li>◦ meeting minutes</li> <li>◦ information links</li> <li>◦ graphics/images</li> </ul> </li> </ul>
Mark Pralat	Mobile Group	✓Go	<p>It worked as expected. I created the mobile group, added meetings etc.</p> <p>Some areas need some refinement:</p> <ul style="list-style-type: none"> <li>• was able to enter incorrect email addresses, e.g. 0@1 was accepted.</li> <li>• long (150 digits) numbers were valid entries for first name and last name <ol style="list-style-type: none"> <li>1. <b>this has been addressed</b> (does someone know a Jira ticket ID?) <b>BUT</b></li> <li>2. until we incorporate a feature to convert into tinyURLs, the new size limits for Website and Calendar URLs are too restrictive – should be at least the 254 we allow for Location (Map URL)</li> </ol> </li> <li>• Adding venue - Mobile Hangout - can only list Busy Body as contact. Cannot unselect.</li> <li>• meeting - need to create venue first, just document venue needs to be created first.</li> </ul>
Ray Bloom	QA Study Group	✓Go	<p>Good work by all involved; sufficiently implemented and functional to release and start using.</p> <ul style="list-style-type: none"> <li>• Change the title element of the initial frontend page to something more meaningful than "TssgDbApp"</li> <li>• All frontend page titles should, perhaps, be something other than, for example, "TSSG Home", as that's not really the actual TSSG home page that people will see when they go to <a href="http://tssg.tech">http://tssg.tech</a></li> <li>• Very long table values (such as the first and last name fields in the users' list) make it almost impossible for a user to scroll all the way to the right to use the Edit or Del buttons on the correct entry. (Perhaps move the buttons to the far left on all tables, make the long fields/columns individually horizontally scrollable, or limit the length of data)</li> <li>• Move the descriptive text for the list pages (e.g., "A list of all venues currently in the database.") to the top of the page, perhaps between the page title and the table headers</li> <li>• We probably shouldn't allow whitespace in a username</li> <li>• Usernames are case-sensitive during login, this is non-standard IMHO</li> <li>• Home page (after user successfully logs in) should show username (either in addition to or instead of user's first and last names)</li> <li>• <a href="#">TWS-340</a> - Getting issue details... <span>STATUS</span> Add copyright in footer</li> <li>• On the "List Team Schedule" page, the on-page title shouldn't be "Select Team Name"</li> <li>• On the "List Team Schedule" page, move "(select one)" hint into the dropdown box as its hint</li> <li>• Add the capability for a user to see which teams they belong to (perhaps just dump it out on the "Current User" page, but it might also be useful for an admin to see all the teams <i>each</i> user is a member of</li> <li>• Add tips for when the user hovers over (with a mouse or whatever) a field name in all pages</li> <li>• Sometimes, when I'm adding data and click Add, I see the pink flash of an error message, but it's gone so fast, I can't capture it</li> <li>• When creating a new team, the specified team lead should be considered to be added as a member so as to not get the "At least one member is required" and/or "One or more team members are required." error</li> </ul>
Jim Turner	Web Team	✓Go	<p>WebTeam approval is a given, our Jira is full of details</p> <ul style="list-style-type: none"> <li>• But we are turning these observations into backlog tickets</li> <li>• and at least Jim has made note(s) on others' observations, to address in these tix</li> </ul>
Ralph A. Navarro Jr.	DevOps Team	✓Go	<ul style="list-style-type: none"> <li>• ★ Pushpa did a PoC for <a href="#">SonarQube</a>; a static code analyzer.</li> <li>• We deployed SonarQube to <a href="http://sq.technologynursery.org">sq.technologynursery.org</a>. The tool initially found over 300 potential vulnerabilities and bugs. These included <a href="#">WCAG2</a> issues.</li> <li>• ★ Paul has fixed all bugs and potential vulnerabilities found by SonarQube.</li> </ul>
Norm Heckman Alan Rawsthorne	Data Analytics Team	✓Go	<p>Approval by Alan on behalf of Norm, but any feedback we have from any Data team members came from discussions in Data Science group, see below</p>
Candy O'Sullivan (Sutherland)	Data Science Learning Team	✓Go	<p>Scheduledb Feedback</p> <ol style="list-style-type: none"> <li>1. <a href="#">TWS-339</a> - Getting issue details... <span>STATUS</span> Suggestion for recurring events.</li> <li>2. <a href="#">TWS-341</a> - Getting issue details... <span>STATUS</span> Suggestion for automatic Zoom invite sent at meeting creation allowing invitees to add zoom meeting for example to their google calendar(or other calendar).</li> <li>3. <a href="#">TWS-342</a> - Getting issue details... <span>STATUS</span> Suggestion for single login i.e. using Facebook or Google account.</li> <li>4. <a href="#">TWS-343</a> - Getting issue details... <span>STATUS</span> Suggestion to display the time zone the meeting times are using.</li> <li>5. <a href="#">TWS-345</a> - Getting issue details... <span>STATUS</span> Suggestion to further define and improve on the phone number entry.</li> </ol>

Lora Bates	TSSG		
------------	------	--	--

## Conclusion

When all of TSSG management has approved these services for production, the following steps will be done:

- ☒ We will follow the guidelines for code review as specified in Requesting a merge to scheduledb of your code contributions.
- ☒ The scheduledb branch will be merged into the master branch. The master branch contains the state of the current tssg.tech web site.
- ☒ The database will be compared to the data on the current site and updated to match
  - ☒ We should split up the List All pages and each Web Team member update his/her collection data vs the True Data
  - ☐ ~~Afterward, do the same for the static pages in the new Site folder.~~ Jim Turner 17 Mar 2021 (or when it is done) Let client make modifications to either the database (through the FrontEnd) or raise tickets for the Web Team to change the Site content as necessary.
  - ☒ ~~Final step: see to it somehow that this (newest) data is in the (initializing) db saved in the Git repository.~~ Not needed because the db is persistent. Just change the data through the FrontEnd UI.
- ☒ Ralph A. Navarro Jr. 13 Mar 2021 Upgrade a13 storage from 128G to 512G
- ☐ A Jenkins project will deploy the master branch to a staging (i.e. QA) endpoints.
  - ☒ decide the names of the services' subdomain endpoints
    - ☒ Website  
https://tssg.tech a13
    - ☒ Backend  
https://backend.tssg.tech a13
    - ☒ Admin (FrontEnd)  
https://admin.tssg.tech a13
  - ☒ Ralph A. Navarro Jr. 13 Mar 2021 configure Nginx to reverse-proxy them to the underlying services
  - ☒ Paul Scheidemantel 14 Mar 2021 Enable scheduledb.env working in private branch taken from master branch to configure either dev or prod endpoints by passing an environment variable (deployEnv) option (default=dev) to scheduledb.env.
  - ☐ Paul Scheidemantel 14 Mar 2021 Create a prod.install for deploying scheduledb to prod.
  - ☒ Ralph A. Navarro Jr. 14 Mar 2021 configure Jenkins
  - ☒ Ralph A. Navarro Jr. 14 Mar 2021 execute the deploy
- ☐ Automated and manual tests will be run against the ~~staging~~ production area. This will verify that the merge has happened successfully before moving to the next step in the pipeline. Corrections will be made if needed until the staging area quality can be assured.
  - ☐ create "smoke" tests to verify service inter-connectivity
  - ☐ enhance code coverage for all UX features
- ☒ Another Jenkins project will deploy the master branch to the production endpoints. We will be
- ☐ Automated and manual tests will be run against the production endpoints. This will verify the quality of the production endpoints. Corrections will be made if needed until the production area quality can be assured.
- ☐ Once in production, create a backup script to backup the database data periodically (either cron job or manually).