

Troubleshooting Common Issues

Performance Problems

- **Large memory usage:** Process voxels in batches rather than loading all into memory
- **Slow execution:** Optimize algorithms and reduce unnecessary computations
- **Timeout errors:** Break large jobs into smaller regions

Data Issues

- **Missing properties:** Always use the `.get()` method with default values
- **Unexpected data types:** Validate and convert data types as needed
- **Empty regions:** Check for zero voxel counts before performing calculations

Runtime Errors

- **Import errors:** Ensure all required libraries are available in the runtime environment
- **Syntax errors:** Test your code locally before uploading
- **Logic errors:** Use logging extensively to trace execution flow

Conclusion

Spatial lambdas bring high-performance analytics directly to your data, enabling custom computations at massivescale. By understanding how to author, configure, and run these serverless functions, you can:

- **Extract deeper insights** from voxel models
- **Automate complex spatial analyses**
- **Process trillions of voxels** efficiently
- **Create custom workflows** tailored to your specific needs

The power of spatial lambdas lies in their ability to execute near your data, eliminating the need to transfer massive datasets and enabling real-time analysis at unprecedented scales.

Next Steps

To advance your spatial lambda expertise:

1. **Explore built-in templates** available in the VoxelSpace platform
2. **Experiment with different processing techniques** and algorithms
3. **Study advanced examples** in the developer documentation
4. **Consider upgrading your plan** for increased compute resources and concurrent processing
5. **Join the VoxelSpace community** to share techniques and learn from other users

For comprehensive developer documentation and advanced tutorials, consult the full manuals within VoxelSpace once you have platform access.

Revision #1

Created 21 October 2025 17:54:52 by admin

Updated 21 October 2025 17:59:27 by admin