Basic / Beginner-Friendly Repositories

1. SETL (Scala ETL framework) — github.com/SETL-Framework/setl
    - A clean, modular Scala ETL framework built on Spark. GitHub
    - Why useful: Good for learning structure (modules, dependencies, transforms).
    - How to use: Clone and walk through sample pipelines, then adapt it for a simple CSV→Parquet→DB flow.
2. spark-etl-framework — github.com/qwshen/spark-etl-framework
    - Pipeline-based data transformation using Spark SQL + Scala. GitHub
    - Why useful: Focused on "end-to-end" from ingestion to transformation.
    - Use case: Good for your intermediate stage where you want to move beyond trivial examples.
3. MyDataFramework — github.com/vbounyasit/MyDataFramework
    - A Scala ETL framework for data engineers. GitHub
    - Why useful: More "framework" oriented, suitable when you want to build reusable pipelines rather than one-off scripts.
    - Use case: As you advance, you might refactor your own ETL apps based on this.
4. etl-spark — github.com/alexland/etl-spark
    - A simpler extract-transform-load pipeline in Scala + Spark. GitHub
    - Why useful: Minimalistic, great for beginners who want a "first real pipeline" to run.
    - Use case: Clone this, run it with your environment, and then extend it (new source, new transform, new sink).

Advanced / Production-Oriented Repositories

1. Teams-League-Airflow-Spark-Scala-ETL — github.com/tosun-si/teams-league-airflow-spark-scala-etl
    - Real-world use case: Cloud Storage + Spark (Dataproc serverless) + Scala + BigQuery, orchestrated by Apache Airflow. GitHub
    - Why useful: Good for seeing how orchestration, cloud, and Spark integrate.
    - Use case: When you're ready to learn end-to-end architecture (ingest → ETL → load → orchestration).
2. spark-etl — github.com/aphp/spark-etl
    - Contains modules around ETL processes in Scala/Spark + PostgreSQL. GitHub
    - Why useful: Mixes Spark with JDBC sinks, good to practice integration with RDBMS.
    - Use case: Load from Postgres, transform, write back or to a data warehouse—this is often needed in real jobs.
3. Scala-and-Spark-in-Practice — github.com/ruslanmv/Scala-and-Spark-in-Practice-

    - A collection of Scala + Spark practice exercises/pipelines. GitHub
    - Why useful: Great for advanced practice, studying patterns, refactoring, performance.
    - Use case: Use this to benchmark your own skills, find performance bottlenecks, refactor code.

How to Use These Repositories for Your Learning & Training

- **Clone each repo**: `git clone <repo-url>`
- **Examine the dependencies** (in `build.sbt` or `pom.xml`) to ensure Scala version (2.12.x) and Spark version (3.x) match your setup.
- **Run the pipeline**: Use `sbt run` or equivalent. Fix environment, data paths, configs.
- **Modify / Extend**: Add a new data source, add a transformation, add a UI or a data sink.
- **Refactor for production**: Add logging, error handling, partitioning, performance tweaks, metrics.
- **Use for interview/training**: Study patterns, code structure, modularization, unit tests.
